

**Universidade Federal do Rio Grande do Norte
Centro de Ciências Exatas e da Terra
Departamento de Informática e Matemática Aplicada
Programa de Pós-Graduação em Sistemas e Computação**

Segmentação Fuzzy de Imagens e Vídeos

Lucas de Melo Oliveira

Natal, Fevereiro de 2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

**Universidade Federal do Rio Grande do Norte
Centro de Ciências Exatas e da Terra
Departamento de Informática e Matemática Aplicada
Programa de Pós-Graduação em Sistemas e Computação**

Segmentação Fuzzy de Imagens e Vídeos

Exame de dissertação submetido ao Programa de Pós-Graduação em Sistemas e Computação do Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte como parte dos requisitos para a obtenção do grau de Mestre em Sistemas e Computação (MSc.).

Lucas de Melo Oliveira

Natal, Fevereiro de 2007

Segmentação Fuzzy de Imagens e Vídeos

Lucas de Melo Oliveira

Este exame de dissertação foi avaliado e considerado aprovado pelo Programa de Pós-Graduação em Sistemas e Computação do Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte.

Prof. Dr. Bruno Motta de Carvalho
Orientador

Profa. Dra. Thaís Vasconcelos Batista
Coordenadora do Programa

Banca Examinadora:

Prof. Dr. Bruno Motta de Carvalho
Presidente

Prof. Dr. Luiz Marcos Garcia Gonçalves

Prof. Dr. Marcelo Ferreira Siqueira

OLIVEIRA, Lucas de Melo

Segmentação Fuzzy de Imagens e Vídeos / Lucas de Melo Oliveira. Natal: 2007.

88p. : xviii.

Orientador: Prof. Dr. Bruno Motta de Carvalho

Dissertação (Mestrado) – Universidade Federal do Rio Grande do Norte. Centro de Ciências Exatas e da Terra. Departamento de Informática e Matemática Aplicada.

1. Inteligência Computacional – Tese. 2. Segmentação Fuzzy de Vídeos e Imagens. 3. Fluxo Óptico. 4 Correlação de Pearson.

"Seja bendito o nome de Deus de eternidade a eternidade, porque dele é a sabedoria e o poder. Ele é quem muda o tempo e as estações, remove reis e estabelece reis; ele dá sabedoria aos sábios e entendimento aos entendidos. Ele revela o profundo e o escondido; conhece o que está nas trevas e com ele mora a luz. A tí, o Deus de meus pais, eu te rendo graças e te louvo..."

Daniel 2:20-23.

AGRADECIMENTOS

Sou grato a Deus por tudo. A Ele a gratidão pela oportunidade de ter finalizado mais uma etapa dos meus estudos. Obrigado Jesus.

Agradeço a meus pais Pedro Ivo e Ana Maria pelo incentivo, confiança e apoio despendido durante toda a minha vida. Vocês foram fundamentais para meu bem estar durante o mestrado. Obrigado também por vocês terem construído a minha família e ter me presenteado com meus amados irmãos Leandro, Liana, Lucio e Bartolomeu. Meus sinceros agradecimentos as minha cunhadas Gleide, Gleides e Poliana. Agradeço também a professora Maria José *Zete* dos Santos pela revisão do texto e pela ajuda durante meus primeiros passos na vida acadêmica.

Agradeço ao professor Dr. Bruno Motta de Carvalho pela orientação, paciência e amizade despendida durante esse dois anos de estudo. Obrigado por me apresentar a uma área antes desconhecida e hoje tão fascinante. Meus agradecimentos também a Simone, Cecília e Carolina pela compreensão e pelo suporte na produção dos vídeos.

Meus agradecimentos ao CNPq pela apoio financeiro concedido durante boa parte do mestrado. Agradeço ao membros e voluntários da equipe do *AnimVideo*, em especial a Gilbran, Isânio e Rafael, pela oportunidade de trabalharmos juntos no desenvolvimento da ferramenta *AVP Rendering* durante esses dois anos.

Meus sinceros agradecimentos a professora Dr. Anne Magaly de Paula Canuto pela recepção, atenção e primeiras orientações no mestrado. Aqueles primeiros meses foram de aprendizados intensos. Agradeço também aos professores Flávio Morais de Assis Silva e Claudete Alves da Universidade Federal da Bahia pela incentivo e apoio à minha decisão de fazer o mestrado.

Agradeço aos professores e funcionários do Departamento de Informática e Matemática Aplicada pelo apoio e por fornecer a infraestrutura necessária para o desenvolvimento desta pesquisa. Aos professores e alunos do Lablic - Laboratório de Inteligência Computacional - meus sinceros agradecimentos pelo apoio, companheirismo e pelo momentos de aprendizado. Aos companheiros de PPgSC pela oportunidade de conviver, aprender e trocar experiências profissionais e de vida durante esse dois anos de estudos. Um agradecimento especial aos amigos Laurindo, Maurício, Cláudia Fernanda, Camila, Demost, Daniel, Cristine e Lígia. Aos companheiros de TO, agradeço ao raros constantes momentos de pesquisa aplicada ao entretenimento coletivo. Aos amigos Raul e Adriana, agradeço pela possibilidade de me fazer sentir em uma família mesmo tão longe de casa.

Ao Instituto Recôncavo de Tecnologia, na pessoa do coordenador Roberto Szabo e dos professores Hugo Saba e Eduardo Jorge, pela oportunidade de ingressar na área da pesquisa aplicada e pela compreensão e incentivo à minha decisão de fazer o mestrado.

Agradeço aos amigos e irmãos da Igreja Batista Missionária de Ponta Negra. Foi um presente de Deus pra minha vida encontrar e conviver com todos vocês todo esse tempo que passei em Natal.

Por fim agradeço a todas as pessoas e instituições que contribuíram para minha formação acadêmica, profissional e pessoal.

Resumo

Segmentação de imagens é o processo que subdivide uma imagem em partes ou objetos de acordo com alguma característica comum. Já na segmentação de vídeos, além dos quadros serem divididos em função de alguma característica, é necessário obter uma coerência temporal entre as segmentações de *frames* sucessivos do vídeo. A segmentação *fuzzy* é uma técnica de segmentação por crescimento de regiões que determina para cada elemento da imagem um grau de pertinência (entre zero e um) indicando a confiança de que esse elemento pertença a um determinado objeto ou região existente na imagem. O presente trabalho apresenta uma aplicação do algoritmo de segmentação fuzzy de imagem, e a extensão deste para segmentar vídeos coloridos. Nesse contexto, os vídeos são tratados como volumes 3D e o crescimento das regiões é realizado usando funções de afinidade que atribuem a cada pixel um valor entre zero e um para indicar o grau de pertinência que esse pixel tem com os objetos segmentados. Para segmentar as seqüências foram utilizadas informações de movimento e de cor, sendo que essa última é proveniente de um modelo de cor convencional, ou através de uma metodologia que utiliza a correlação de Pearson para selecionar os melhores canais para realizar a segmentação. A informação de movimento foi extraída através do cálculo do fluxo óptico entre dois *frames* adjacentes. Por último é apresentada uma análise do comportamento do algoritmo na segmentação de seis vídeos e um exemplo de uma aplicação que utiliza os mapas de segmentação para realizar renderizações que não sejam foto realísticas.

Área de Concentração: Inteligência Computacional

Palavras-chave: Segmentação Fuzzy, Fluxo Óptico, Processamento de Imagens, Processamento de Vídeos

Abstract

Image segmentation is the process of subdividing an image into constituent regions or objects that have similar features. In video segmentation, more than subdividing the frames in object that have similar features, there is a consistency requirement among segmentations of successive frames of the video. Fuzzy segmentation is a region growing technique that assigns to each element in an image (which may have been corrupted by noise and/or shading) a grade of membership between 0 and 1 to an object. In this work we present an application that uses a fuzzy segmentation algorithm to identify and select particles in micrographs and an extension of the algorithm to perform video segmentation. Here, we treat a video shot is treated as a three-dimensional volume with different z slices being occupied by different frames of the video shot. The volume is interactively segmented based on selected seed elements, that will determine the affinity functions based on their motion and color properties. The color information can be extracted from a specific color space or from three channels of a set of color models that are selected based on the correlation of the information from all channels. The motion information is provided into the form of dense optical flows maps. Finally, segmentation of real and synthetic videos and their application in a non-photorealistic rendering (NPR) toll are presented.

Area of Concentration: Theory and Computational Intelligence

Key words: Fuzzy Segmentation, Optic Flow, Image Processing, Video Segmentation, Image Segmentation.

Sumário

Lista de Símbolos e Abreviaturas	i
1 Introdução	1
1.1 Motivação e Objetivos	2
1.2 Contribuições	2
1.3 Organização do Trabalho	4
2 Segmentação de Imagens	5
2.1 Segmentação de Imagens Monocromáticas	6
2.1.1 Segmentação Baseada em Regiões	8
2.2 Segmentação de Imagens Coloridas	10
2.2.1 Modelos de Cores	11
2.2.2 Seleção de Informação de Cores	14
2.2.3 Técnicas de Segmentação de Imagens Coloridas	17
3 Segmentação de Vídeos e Estimação de Movimento	19
3.1 Introdução	19
3.2 Técnicas para Segmentação de Vídeos	20
3.3 Estimação de Movimento	22
3.4 Fluxo Óptico	24

4	Segmentação Fuzzy de Imagens	30
4.1	Introdução	30
4.2	Segmentação Fuzzy Simultânea de Múltiplos objetos	31
4.3	Algoritmo para Segmentação Fuzzy	34
4.4	Algoritmo Rápido para Segmentação Fuzzy	37
4.5	Exemplos de Segmentação Fuzzy de Imagens	39
4.5.1	Seleção de Partículas em Micrografos	41
5	Segmentação Fuzzy de Vídeos	47
5.1	Segmentação Fuzzy de Vídeo por Propagação de Sementes	47
5.2	Segmentação Fuzzy de Volumes de Vídeo	51
5.3	Seleção de Informações de Cores Não-Correlacionadas	56
5.4	Implementação	59
5.5	Experimentos	62
5.6	Uma Aplicação da segmentação Fuzzy para renderizações NPR	76
6	Conclusão	79

Lista de Figuras

2.1	Exemplo de digitalização de imagens.	6
2.2	Segmentação utilizando <i>Thresholding</i> . Uma imagem de ressonância magnética de um crânio (a) e a sua segmentação por <i>thresholding</i> (b).	7
2.3	Vizinhanças de pixel. É geralmente utilizadas na segmentação por crescimento de regiões, onde o pixel do centro é pixel p e os seus vizinhos de quatro (a), diagonal (b) e de oito (c) estão pintados de cinza.	9
2.4	Representação do modelo <i>RGB</i> por um cubo. O eixo X , Y e Z correspondem respectivamente aos canais vermelho (<i>red</i>), verde (<i>green</i>) e azul (<i>blue</i>) (Figura extraída de wikipedia.org).	11
2.5	Análise isolada dos canais dos modelos <i>RGB</i> (b-d), YC_bC_r (e-g) e <i>HSI</i> (h-j) em uma imagem colorida (a).	15
3.1	Estimação de movimento. O deslocamento do pixel p (de p_1 para p_2) entre os instantes t e $t+1$	23
3.2	Equação de restrição do fluxo óptico. A reta produzida pela equação de restrição do fluxo óptico e a linha de restrição (pontilhada).	25
3.3	Estimação do movimento com diferentes parâmetros. Dois <i>frames</i> consecutivos de um vídeo (a) e (b) e a estimação de movimento na sub-imagem destacada em (a) e (b) utilizando o algoritmo de Proesmans com os seguintes parâmetros: (c) lambda 10, 5 níveis e 15 iterações; (d) lambda 1000, 5 níveis e 15 iterações; (e) lambda 1000, 2 níveis e 15 iterações.	27
3.4	Exemplo da estimação de movimento. Dois <i>frames</i> de uma seqüência (a) (b) e o mapa do fluxo óptico (c) calculado através de uma versão multi-resolução do algoritmo de Proesmans [PGPO94]. Foram utilizados como parâmetros lambda igual a 1000, 15 iterações e 4 níveis.	29

4.1	Exemplo da segmentação por <i>Thresholding</i> . Uma figura com seu real valor de intensidade (a) e uma imagem binária produzida por um limiar (b) (Figura extraída de Carvalho [CGHK99]).	31
4.2	Exemplo de duas correntes existentes em uma imagem	33
4.3	Ilustração gráfica do Teorema 1.	35
4.4	Comparação do algoritmo MOSFS original e a versão rápida. Duas fatias de um volume de tomografia computadorizada (a-b) e as suas correspondentes 4-segmentações obtidas utilizando o algoritmo original MOFS (c-d) e versão rápida do MOFS (e-f) (Imagens obtidas de Carvalho et al. [CHK05]).	40
4.5	Exemplo de segmentação fuzzy. Uma imagem de ressonância magnética de um crânio (a) e a 4-segmentação deste crânio (b) (Imagens obtidas de Carvalho et al. [CHK05]).	41
4.6	Sub-imagem de um micrografo. Uma sub-imagem (200×200 pixels) retirada de um micrografo obtido através de um microscópio eletrônico (imagem retirada de Carvalho et al. [COG06])	42
4.7	Fluxograma detalhado do funcionamento do procedimento para seleção de partículas em micrografos.	43
4.8	Imagens do processo de identificação e seleção de partículas no micrografo. Sub-imagem de um micrografo original (a), imagem pré-segmentada ou mapa de afinidade (b), threshold do mapa de afinidade (c), centro das partículas selecionadas por correlação (d), o micrografo depois da etapa do refinamento (e) e o micrografo com a seleção manual (f). (Imagens extraídas de [COG06]).	45
5.1	Funcionamento da segmentação fuzzy por propagação de sementes. Depois do primeiro frame ser segmentado, no Passo 1, a semente é propagada no Passo 2, e então o próximo frame é segmentado (Passo 3).	49
5.2	Segmentação fuzzy por propagação de semente. O primeiro e o sexto frame de um vídeo (original (a-c) e segmentado (b-d))	50
5.3	Dificuldades na segmentação fuzzy por propagação de semente. Devido a alguns pixels se localizarem próximo às bordas, existe uma perda de precisão na segmentação (horizontalmente, verticalmente e em profundidade).	51

5.4	Um vídeo visto como um volume. O volume 3D (a) e as direções para onde as regiões podem ser expandidas (b).	52
5.5	Vizinhança de face 3D. Voxel central (cinza escuro) e seus vizinhos adjacentes de face (cinza claro).	52
5.6	Exemplo de segmentação fuzzy de vídeo. Frames originais (a,c,e) e a segmentação fuzzy (b,d,f) de um vídeo. Cada linha corresponde a um intervalo de 15 frames.	53
5.7	Utilização da informação de movimento para diferenciar o mouse do background. Frames originais (a-b), a sua segmentação fuzzy utilizando somente informações de cor (c-d) e com informações de cor e movimento (e-f).	55
5.8	A imagem original do <i>Taz</i> (a) e a visualização isolada dos canais R (b), G(d), B(f), H (c), S (e) e I (g).	57
5.9	Seleção dos canais de cores utilizando a correlação de Pearson. As setas indicam as menores correlações achadas entre os canais.	59
5.10	Fluxograma de funcionamento da segmentação fuzzy no <i>AVP Rendering</i>	59
5.11	Tela do <i>AVP Rendering</i> . Selecionando as sementes para segmentar um vídeo	60
5.12	Tela do <i>AVP Rendering</i> : Opção onde o usuário poderá segmentar o vídeo.	61
5.13	Fluxo Óptico gerado por três frames (a-c) do vídeo <i>Traffic Car</i> e os seus mapas do fluxo óptico.	65
5.14	Resultado das segmentações do vídeo <i>Traffic Car</i> variando os pesos aplicados às informações e cor e movimento, usando os pesos descritos na Tabela 5.2.	66
5.15	Resultado da segmentação do vídeo <i>Traffic Car</i> utilizando o modelo <i>RGB</i> . Cada linha corresponde a um intervalo de 20 frames.	68
5.16	Resultados das segmentações do vídeo <i>Taz</i> , mostrando os frames 1, 30 e 70 segmentados com os canais I_3BH (d-f) selecionado pela metodologia da correlação de Pearson e pelo modelos de cores <i>RGB</i> (g-i), <i>HSI</i> (j-l) e os canais $I_1I_2I_3$ (m-o).	69
5.17	Primeiro frame da seqüência sintética original (a) e com a adição de ruídos (b).	70

5.18	Segmentação da seqüência original do <i>Office</i> . Frame 5 e 15 segmentados utilizando o modelo $I_1I_2I_3$ (a-b) e com os canais HI_3 (c-d).	72
5.19	Segmentação da seqüência <i>Office</i> com ruído. Frame 5 e 15 segmentados utilizando o modelo $I_1I_2I_3$ (a-b) e com os canais I_2I_1S (c-d).	73
5.20	Resultado da segmentação da seqüência do <i>Demost</i> . Os frames 1, 7, 14 e 21 originais (a,c,e,g) e segmentados com o modelo RGB (b-d-f-h).	74
5.21	Resultado da segmentação da seqüência da <i>Caneca</i> com o modelo de cor $I_1I_2I_3$. Cada linha corresponde a um intervalo de 15 frames.	75
5.22	Frame original (a) e o mapa de segmentação (b) do primeiro frame, e a Renderização NPR <i>Csand</i> em dois frames da seqüência. O primeiro renderizando só o sapo (c) e o segundo só a barriga e joelho do sapo (d). .	77
5.23	Exemplo da aplicação da segmentação fuzzy para renderização NPR. Três frames utilizando a técnica de pintura impressionista.	78

Lista de Tabelas

4.1	Quantidade de falsos positivos e falsos negativos para seleção de partículas em micrografos.	46
5.1	Características dos vídeos utilizados nos experimentos.	63
5.2	Configuração para segmentação do vídeo <i>Taz</i> para mostrar a influência da informação de cor.	64
5.3	Configuração para segmentação do vídeo <i>Taz</i> para mostrar a influência da informação de cor.	67
5.4	Objetos segmentados no vídeo do <i>office</i>	70
5.5	Resultados das segmentações do vídeo sintético sem a adição de ruídos, onde os melhores resultados para cada seleção estão em negrito.	71
5.6	Resultados das segmentações do vídeo sintético com a adição de ruídos, onde os melhores resultados para cada estão em negrito.	71
5.7	Tempo da segmentação fuzzy para os vídeos analisados.	73

Lista de Símbolos e Abreviaturas

σ Função que mapeia cada spels c do conjunto V em um vetor de $M+$ dimensões σ^c .

$(c^{(k-1)}, c^{(k)})$ Uma link entre os spels $k - 1$ e k .

$FD_{n,n-1}(x, y)$ Função que calcula a diferença entre os valores atribuídos aos pixels de dois frames adjacente.

M Quantidade de objetos segmentados.

V Conjunto de pixels sementes.

$Z_{n,n-1}(x, y)$ Função que indica se uma semente deverá ser propagada para o próximo frame.

$\Omega_{i,j}$ Calcula a correlação de Pearson entre as variáveis i e j .

Ψ O conjunto com M funções de afinidade.

Υ^i Calcula o valor da menor correlação de Pearson entre a variável i as outras $(k - 1)$ variáveis.

\bar{u} Vetor do fluxo óptico relativo ao deslocamento vertical.

\bar{v} Vetor do fluxo óptico relativo ao deslocamento horizontal.

$\langle c^{(0)}, \dots, c^{(K)} \rangle$ Uma corrente de K spels entre os spels $c^{(0)}$ e $c^{(k)}$.

\mathcal{V} O conjunto com M conjuntos de pixels sementes V_m ($1 \leq m \leq M$).

$\mu_{\sigma,m,W}(c)$ Denota a ψ_m -força máxima de uma σm -corrente de um *spel* em W para c .

$\phi_{\Psi}(c, d)$ Mínimo valor de $c, d \in V$, para todo para todo $c, d \in V$, entre todos os m objetos ($1 \leq m \leq M$).

$\psi_m(c, d)_{cor}$ Função parcial de afinidade relativa a informação de cor.

$\psi_m(c, d)_{vel}$ Função parcial de afinidade relativa a informação de movimento.

- ψ Função de afinidade que atribui um número real positivo entre 0 e 1 para cada par ordenado de *spels*.
- ψ -força Força de um link entre dois *spels* definida por uma função de afinidade ψ .
- $\psi_m(c, d)$ Valor da função de afinidade entre os *spels* c e d .
- $\rho_{r,s}(x)$ Função de densidade de probabilidade gaussiana com média r e desvio padrão s multiplicada por uma constante para que o seu valor máximo seja 1.
- σ -corrente Corrente que $\sigma_m^{c(k)} > 0$, para $0 \leq k \leq K$.
- σ^c Vetor de $M+1$ dimensões que armazenas as afinidades do *spel* c para os m objetos.
- σ_0^c Valor da maior afinidade entre o *spel* c e o m -ésimo objeto ($1 \leq m \leq M$).
- σ_m^c Valor da afinidade do *spel* c para o objeto m .
- a Diferença absoluta entre as intensidades dos *spels* c e d .
- a_m Média para todas as diferenças absolutas de intensidade entre os *spels* em V_m e seus vizinhos de borda.
- b_m Desvio padrão para todas as diferenças absolutas de intensidade entre os *spels* em V_m e seus vizinhos de borda.
- $c(x, y)$ Calcula a correlação entre as imagens $f(x, y)$ e $w(x, y)$ no ponto (x, y) .
- $c_L(x, y)$ Calcula a correlação local as imagens $f(x, y)$ e $w(x, y)$ no ponto (x, y) .
- $c_N(x, y)$ Calcula a correlação normalizadas entre as imagens $f(x, y)$ e $w(x, y)$ no ponto (x, y) .
- g Média entre as intensidades dos *spels* c e d .
- g_m Média para a média de todos pares formados entre os *spels* em V_m e os seus vizinhos de borda.
- h_m Desvio padrão para a média de todos pares formados entre os *spels* em V_m e os seus vizinhos de borda.
- s_n^c Define o valor da maior força do objeto n ($1 \leq n \leq M$) que parte de um pixel semente de V_n e passa por c .

Capítulo 1

Introdução

O processamento digital de imagens tem como alguns de seus objetivos o melhoramento da informação visual para a interpretação humana, o processamento de imagens para o seu armazenamento, transmissão, representação e utilização em sistemas computacionais. A segmentação de imagens é uma das etapas desse processamento onde a imagem é dividida em partes ou objetos de acordo com algum critério de homogeneidade [GW02]. A partir de uma imagem segmentada, ou seja, dividida em objetos, ela pode ser utilizada em aplicações que envolvem a detecção e identificação de objetos, automação industrial, análise de superfícies, dentre outras.

O processo de segmentação um objeto de um background de uma imagem é uma tarefa difícil para um computador. Algumas características como a existência de ruídos, iluminação não uniforme e textura tornam ainda mais complicada a tarefa da segmentação pois podem inserir nas imagens informações que dificultam a separação dos objetos. A segmentação *fuzzy* de imagens é uma técnica que determina para cada elemento da imagem um grau de pertinência que indica a certeza desse elemento pertencer ou não a um determinado objeto existente na imagem [CHK05]. Essa técnica vem obtendo bons resultados na segmentação de imagens com essas características.

A segmentação de vídeos, que pode ser vista como seqüências de quadros ou imagens, consiste na identificação de regiões que são homogêneas de acordo com algum critério pré-definido. Essas regiões podem ser objetos existentes na cena, como um carro ou um motorista, ou o background da cena como uma rodovia. As características mais utilizadas na definição do critério de homogeneidade a ser aplicado na segmentação de vídeos são a intensidade, a cor, a textura e o movimento dos objetos existentes na cena.

Apesar de um vídeo poder ser considerado como uma seqüência de imagens bidimensionais, segmentar um vídeo é diferente de segmentar uma imagem isolada [Bov05]. Isso se deve, dentre outras coisas, a necessidade de se observar a coerência temporal entre os

quadros do vídeo. Além da segmentação de vídeo compartilhar algumas dificuldades da segmentação de imagens, como a existência de ruídos e/ou iluminação não uniforme, a segmentação de vídeos tem suas próprias dificuldades: existência de movimentos bruscos, objetos deformáveis, dentre outras.

1.1 Motivação e Objetivos

A segmentação de vídeos está presente em muitas aplicações que envolvem o processamento de vídeos. Ela pode ser utilizada em aplicações como a recuperação e o reconhecimento de informações existentes em vídeos [ZC97, Bov05], na criação de animações e renderizações não realistas [CBNO06, WXSC04], em sistemas para compressão de vídeos [Che93], dentre outros. Muito esforço tem sido feito para desenvolver técnicas de segmentação de vídeo com a intenção de superar as dificuldades inerentes à segmentação de vídeo e atender às novas necessidades dos sistemas que utilizam a segmentação em uma de suas etapas.

Em várias dessas aplicações, devido às características da cena e até mesmo a qualidade da câmera ou iluminação, existe a necessidade da utilização de algoritmos de segmentação que sejam tolerantes a vídeos corrompidos pela existência de ruídos. Conforme foi discutido em Herman e Carvalho [HC01, CHK05], o algoritmo de segmentação *fuzzy* (*MOFS - Mult-Object Fuzzy Segmentation*) mostrou-se bastante robusto na segmentação de imagens ruidosas. O presente trabalho propõe a adaptação deste algoritmo para segmentar vídeo, explorando a utilização de informações de cor e movimento.

Os objetivos principais deste trabalho são a investigação e a aplicação do método de segmentação *fuzzy* de imagens desenvolvido por Herman e Carvalho [HC01] para novos problemas em segmentação de imagens, além de sua extensão para a segmentação vídeos, e o desenvolvimento de uma metodologia que utilize informações de cor e movimento para segmentar os objetos existentes no vídeo. Um objetivo adicional é o de atribuir importâncias às informações de cor e movimento de uma seqüência de vídeo, com o objetivo de adaptar o algoritmo de segmentação às características específicas do vídeo sendo processado.

1.2 Contribuições

As investigações existentes nesse presente trabalho tiveram como objetivo analisar e aplicar o algoritmo de segmentação fuzzy de imagens e desenvolver uma metodologia para

segmentação fuzzy de vídeos. Uma das primeiras etapas dessa investigação foi o levantamento bibliográfico sobre a segmentação fuzzy de imagens e a aplicação dessa técnica de segmentação para auxiliar na resolução de um problema de processamento de imagens.

Neste trabalho foi desenvolvida uma metodologia para identificação e a seleção de partículas nos micrografos gerados pela Microscopia Eletrônica (EM) e que serão em outra etapa utilizadas na reconstrução 3D destas partículas. As dificuldades desse processo de identificação e seleção se dão devido a alta taxa de ruídos existentes nos micrografos, o tamanho de cada partícula (10 a 15 Å) e a necessidade de seleção de muitas partículas (geralmente, mais do que 10.000 com resolução média de 3 Å) para a reconstrução de um modelo 3D [FWM⁺02, vHGM⁺00, ZMSH03]. A segmentação fuzzy foi utilizada para remover as partes do micrografos com menor probabilidade de existir um partícula.

Os resultados encontrados com a utilização dessa metodologia mostraram-se bastantes satisfatórios. Taxas consideradas razoáveis para os métodos automático ou semi-automático de identificação e seleção de partícula são de 10% para falsos positivos e 25% para falsos negativos. Quanto menor for a taxa de falsos positivos melhor será a construção da macromolécula. Com a utilização da metodologia proposta obtivemos 5,55 % de Falsos Positivos e 23,23 % de Falsos Negativos [COG06].

Outra investigação realizada nesse presente trabalho foi a aplicação do algoritmos MOFS para segmentação de vídeos. Para isto foi desenvolvida uma metodologia semi-automática para segmentar vídeos coloridos como se fossem volumes 3D. Nessa metodologia o usuário indica os pixels sementes para cada objeto a ser segmentado e a partir de informações de cores proveniente do modelo YUV a segmentação é realizada [COA06]. Essa metodologia também foi utilizada para auxiliar na aplicação de renderizações realistas. Para isto os vídeos são segmentados e posteriormente a renderização é aplicada somente nos objetos desejados [CBNO06].

Por fim, foi realizada uma expansão da metodologia mencionada acima para se utilizar informações de cores e movimento para segmentar vídeos. Essa expansão foi realizada com o propósito de aumentar a precisão e robustez nas segmentações. Uma outra característica dessa metodologia é a possibilidade de se realizar análise nos canais de vários modelos de cores (RGB, $I_1I_2I_3$ e HSI) para selecionar três canais de cores para realizar a segmentação com o objetivo melhorar a segmentação. Os resultados das segmentações obtidas nos experimentos, em sua maioria, mostraram-se robustos e consistentes ao longo dos *frames* (quadros) segmentados [CNO06].

1.3 Organização do Trabalho

O Capítulo 2 descreve brevemente alguns dos principais métodos de segmentação de imagens monocromáticas e coloridas, e faz uma rápida apresentação e comparação dos modelos de cores que podem ser utilizados na segmentação. No Capítulo 3 é apresentada uma descrição dos métodos de segmentação de vídeos e é feita uma abordagem nos métodos de estimação de movimento em vídeos, principalmente nos métodos baseados na equação de restrição do fluxo óptico. O capítulo 4 apresenta o algoritmo de segmentação *fuzzy* por crescimento de regiões desenvolvido por Herman e Carvalho [HC01], suas principais características e dois exemplos de aplicações desse método na segmentação de imagens.

Já no Capítulo 5 são apresentadas duas abordagens para segmentar vídeos utilizando o algoritmo de segmentação fuzzy: uma baseada na propagação de sementes; e outra na segmentação de volumes 3D. Nesse capítulo também é realizada uma análise nos resultados obtidos utilizando a segmentação fuzzy para vídeos e apresentada uma aplicação que utiliza a segmentação fuzzy na renderização não realista de vídeos. No Capítulo 6 é apresentada a conclusão, as contribuições desse trabalho e os trabalhos futuros.

Capítulo 2

Segmentação de Imagens

A segmentação de imagens subdivide uma imagem em suas partes ou objetos constituintes [GW02]. Quando o olho humano observa uma cena, a mente humana divide essa cena em objetos, que podem ser uma simples cadeira em um escritório ou até um objeto mais complexo visualmente como uma árvore em uma floresta. Se uma imagem é dividida em partes ou regiões, onde os pixels destas regiões compartilham algumas características comuns (como intensidade, cor ou textura), aplicações como reconhecimento de padrões, compressão e indexação poderão ser realizados [Bov05].

A quantidade de objetos resultantes da segmentação depende do nível de abstração ou detalhamento que se deseja ter da cena observada. Por exemplo, quando se está observando a cena de uma floresta, um primeiro nível de abstração pode resultar somente nos objetos correspondentes aos troncos e copas de árvores. Uma observação mais detalhada da cena pode gerar objetos adicionais como galhos, folhas e animais. Logo, o problema da segmentação não possui uma única solução correta, sendo esta dependente do objetivo desejado.

Uma imagem digital é uma discretização de um objeto ou cena real, que foi capturada por um sensor de algum dispositivo, como uma câmera digital ou um *scanner* de raios X. Nesse processo de discretização, também conhecido como digitalização, uma função contínua $f(x, y)$ é transformada em uma matriz de números inteiros ou de ponto flutuante. A Figura 2.1 ilustra esse processo de digitalização de imagens.

A segmentação de imagens é geralmente um dos primeiros passos em um sistema de processamento de imagens [PP93]. Ela tem um importante papel em um grande número de aplicações como sensoriamento remoto, processamento de imagens médicas, renderização, compressão de vídeos, automação industrial, dentre tantas outras aplicações que envolvem técnicas de processamento de imagens. Geralmente, a qualidade dos resultados

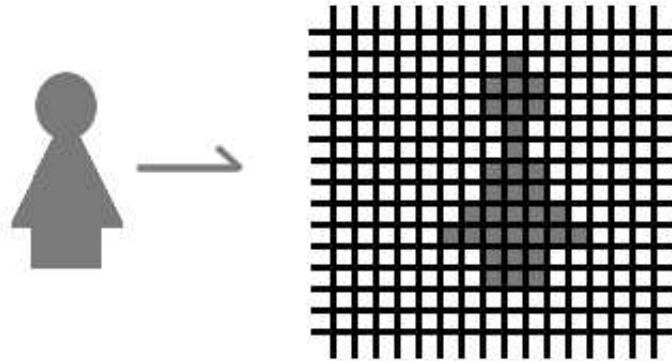


Figura 2.1: Exemplo de digitalização de imagens.

obtidos em tais sistemas depende da qualidade da segmentação, sendo a mesma fortemente relacionada à escolha das características que serão utilizadas para separar os objetos, à confiabilidade da extração dessas características e ao critério utilizado para agrupar os elementos da imagem.

2.1 Segmentação de Imagens Monocromáticas

Imagens monocromáticas são imagens que possuem como único atributo a intensidade. As imagens em tons de cinza, como também são conhecidas as imagens monocromáticas, têm sua intensidade variando de preto que é geralmente a menor intensidade até o branco que é geralmente a maior intensidade. A diferença entre imagens em tons de cinza e imagens preto e branco é que as últimas podem ser consideradas como imagens em tons de cinza que possuem somente dois tons: preto e branco [GW02].

Os algoritmos para segmentação de imagens monocromáticas são baseados em uma das duas propriedades básicas das intensidades das imagens monocromáticas: descontinuidade ou similaridade [GW02]. Na primeira propriedade, a imagem é subdividida detectando-se mudanças abruptas de intensidade, como bordas em uma imagem. Já as principais técnicas baseadas em similaridade particionam as imagens em regiões que são semelhantes de acordo com um conjunto de critérios previamente definidos.

Existem três tipos básicos de descontinuidade: pontos, linhas e bordas. A maneira mais comum de encontrar essas descontinuidades nas imagens é através da aplicação de um filtro no domínio espacial [GW02]. Nesse procedimento, uma janela $N \times N$ (também chamada de filtro, onde cada posição deste contém um coeficiente) é aplicada posicionando o centro desta sobre cada pixel da imagem. Então, é atribuído a esse pixel o valor

da soma dos produtos entre as intensidades dos pixels com os respectivos coeficientes da janela que estão sobre os mesmos.

A técnica de *thresholding* (também conhecida como limiarização) é uma das técnicas mais antigas, simples e utilizadas na segmentação de imagens [PP93]. Essa técnica é baseada na similaridade dos pixels e um limiar t é utilizado para classificar a qual região (objeto ou *background*) um pixel deverá pertencer. Para definir qual o valor do limiar t a ser utilizado, pode ser utilizada informação global da imagem através do histograma das intensidades das imagens monocromáticas, ou informação local somente de uma parte da imagem.

Para cada n valores de limiar, a imagem é dividida em $n + 1$ regiões. Por exemplo, para uma segmentação por *thresholding* com apenas um limiar t , uma imagem será dividida em duas regiões, A e B , utilizando os seguintes critérios:

$$S(x, y) \leq t \Rightarrow (x, y) \in A, \quad (2.1)$$

$$S(x, y) > t \Rightarrow (x, y) \in B, \quad (2.2)$$

sendo $S(x, y)$ a intensidade do pixel na posição (x, y) . Um exemplo de segmentação usando *thresholding* pode ser visto na Figura 2.2.

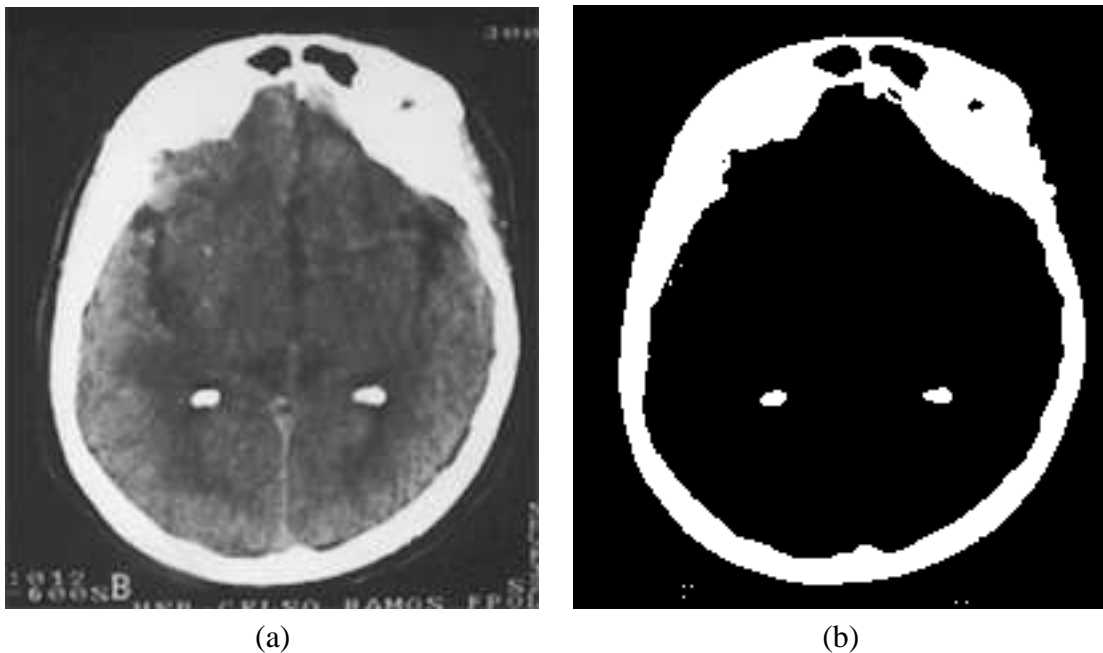


Figura 2.2: Segmentação utilizando *Thresholding*. Uma imagem de ressonância magnética de um crânio (a) e a sua segmentação por *thresholding* (b).

2.1.1 Segmentação Baseada em Regiões

A definição clássica de segmentação baseada em regiões apresentada por Gonzalez [GW02] é a seguinte:

Seja R a representação de uma imagem. A segmentação será considerada o processo de particionar R em n sub-regiões, R_1, R_2, \dots, R_n , tal que

- (a) $\bigcup_{i=1}^n R_i = R$;
- (b) R_i é uma região conectada, $i = 1, 2, \dots, n$;
- (c) $R_i \cap R_j = \emptyset$ para todo i e j , $i \neq j$;
- (d) $P(R_i) = \text{TRUE}$ para $i = 1, 2, \dots, n$;
- (e) $P(R_i \cup R_j) = \text{FALSE}$ para qualquer região que seja *vizinha* das regiões R_i e R_j ,

onde $P(R_k)$ é um predicado lógico definido sobre todos os pontos do grupo R_k e \emptyset é o conjunto vazio.

A condição (a) indica que a segmentação deve ser completa, isto é, todos os pontos da imagem devem pertencer a alguma região. A condição (b) requer que os pontos de uma região devem estar conectados de acordo com algum tipo de conexão pré-definida. A condição (c) implica que as regiões devem ser disjuntas, enquanto que a condição (d) trata da propriedade que um ponto deve satisfazer para pertencer a uma determinada região. Finalmente, a condição (e) indica que as regiões R_i e R_j são diferentes de acordo com a propriedade P .

A segmentação baseada em regiões pode ser feita através de técnicas de divisão de regiões, união de regiões, crescimento de regiões, ou com combinações das anteriores. Na divisão de regiões, o processo de segmentação inicia-se considerando as imagens como uma única região. Se a região não é homogênea, ela é dividida em sub-regiões. Este processo de subdivisão é repetido até que as sub-regiões sejam homogêneas de acordo com um critério que deve ser previamente definido. A segmentação por união de regiões trabalha de modo contrário da divisão de regiões, iniciando com cada pixel sendo uma região e unindo regiões similares até que mais nenhuma união possa ser efetuada baseando-se em um critério pré-definido.

Crescimento de regiões é o procedimento em que os grupos de pixels são formados a partir de um conjunto de pontos chamados de sementes [GW02]. No início do processo,

cada região será composta apenas por um ou mais pixels sementes. Então, os vizinhos que tiverem propriedades similares às sementes desta região serão adicionados a esta região. Existem algumas questões importantes no processo de segmentação por crescimento de regiões, como a escolha do critério de similaridade, a vizinhança a ser utilizada e as sementes que irão influenciar seu funcionamento.

A seleção do critério de similaridade depende não somente da natureza do problema, mas também do tipo de imagem a ser segmentada, pois cada tipo de imagem possui características que influenciam na escolha do critério de similaridade, como na segmentação de imagens monocromáticas, coloridas ou de profundidade (range image) [GW02].

A escolha da vizinhança influenciará diretamente na expansão das regiões, pois a análise dos pixels para uma possível expansão de uma região é feita primeiramente nos pixels vizinhos. Os principais tipos de vizinhanças aplicados no processamento digital de imagem são a vizinhança de quatro (ou de aresta), vizinhança D (diagonal) e a vizinhança de oito, que podem ser vistas na Figura 2.3.

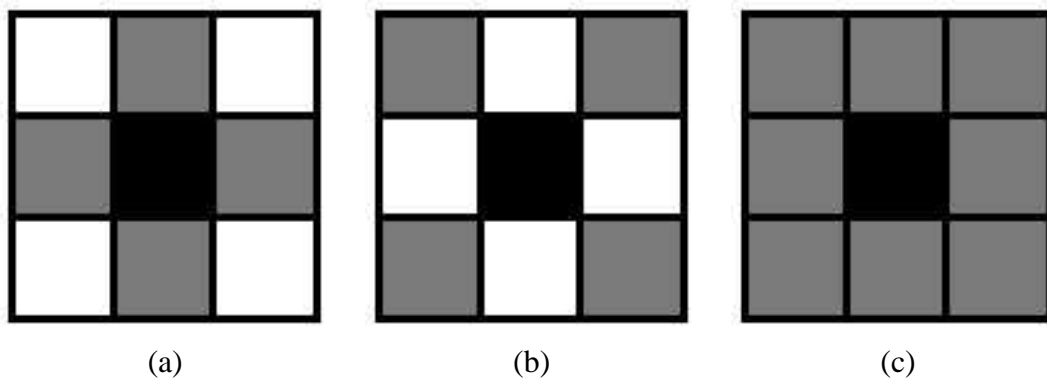


Figura 2.3: Vizinhanças de pixel. É geralmente utilizadas na segmentação por crescimento de regiões, onde o pixel do centro é pixel p e os seus vizinhos de quatro (a), diagonal (b) e de oito (c) estão pintados de cinza.

A escolha dos conjuntos de um ou mais pontos sementes para cada região é uma parte fundamental na segmentação por crescimento de regiões. A seleção das sementes está diretamente relacionada ao conhecimento que o usuário ou programa tem sobre o conteúdo da imagem, e também podem influenciar diretamente no cálculo das propriedades que serão utilizadas no processo de expansão das regiões.

2.2 Segmentação de Imagens Coloridas

A percepção de cores é muito importante para a compreensão de objetos e cenas devido a sua grande capacidade descritiva. Embora o processo seguido pelo cérebro humano na percepção de cores seja um fenômeno fisiopsicológico, que ainda não é completamente compreendido, a natureza física das cores pode ser expressa numa base formal suportada por resultados experimentais e teóricos [GW02]. Muitas aplicações na área de processamento e análise de imagens necessitam trabalhar com objetos e ambientes que contêm informações de cor em suas representações.

A cor é relacionada aos diferentes comprimentos de onda do espectro eletromagnético, sendo que a cor de um material é determinada pelo comprimento de onda dos raios luminosos que suas moléculas constituintes refletem [GW02]. Um objeto terá determinada cor se não absorver justamente os raios correspondentes à frequência de onda desta cor.

De acordo com a teoria dos triestímulos [WS82], a cor pode ser representada por três componentes que são obtidos através da utilização dos filtros de cor $S_x(\lambda)$ (referentes aos três diferentes tipos de foto receptores, chamados de cones, existentes na retina), onde x é igual a R, G ou B e λ é o comprimento de onda. Esses filtros são aplicados ao espectro de radiação $E(\lambda)$ e os três componentes podem ser obtidas através das equações:

$$R = \int_{\lambda} E(\lambda) S_R(\lambda) d\lambda, \quad (2.3)$$

$$G = \int_{\lambda} E(\lambda) S_G(\lambda) d\lambda, \quad (2.4)$$

$$B = \int_{\lambda} E(\lambda) S_B(\lambda) d\lambda, \quad (2.5)$$

As cores R (*red*), G (*green*) e B (*blue*) são usualmente conhecidas como as três cores primárias. Essas três cores correspondem aos três componentes que formam um dos modelos de cores mais conhecido, o modelo RGB . Este modelo, juntamente com os demais modelos de cores existentes, possui particularidades e utilidades que justificam a sua utilização. Por exemplo, temos a utilização do modelo YUV que é utilizado para transmissão de sinais de vídeo e o modelo $CMYK$ que é a base do processo de impressão em quatro cores (quadricomia).

2.2.1 Modelos de Cores

Um modelo de cores, conforme é definido por Gonzalez [GW02], é uma especificação de um sistema de coordenadas e um subespaço de cores no qual cada cor é representada como um único ponto. O espaço RGB é um modelo, ou sistema de representação de cores mais comum nos sistemas de televisão e monitores. Nesse modelo, as cores são formadas variando-se a intensidade nas componentes (também chamado de canais) R , G e B . Uma representação para esse modelo pode ser observada na Figura 2.4, onde cada eixo de um cubo corresponde a um dos canais R , G ou B .

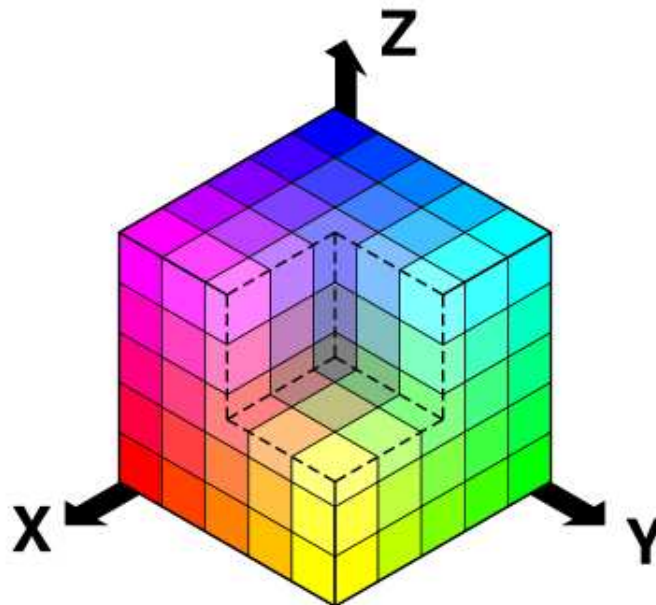


Figura 2.4: Representação do modelo RGB por um cubo. O eixo X , Y e Z correspondem respectivamente aos canais vermelho (*red*), verde (*green*) e azul (*blue*) (Figura extraída de wikipedia.org).

O RGB é um modelo muito utilizado para monitores coloridos, mas não é muito adequado para segmentação de imagens coloridas devido à alta correlação existente entre os seus canais, ou seja, quando ocorre uma variação na intensidade da imagem, ocorre uma variação em todos os três canais do modelo [CM97]. Uma outra característica do modelo RGB é que ele não é perceptualmente uniforme, ou seja, duas cores que são visualmente próximas podem estar mais separadas no cubo que representa o modelo RGB do que duas outras cores que são visualmente mais distantes. Essa característica do modelo RGB faz com que a medida de similaridade obtida através da distância entre dois pontos no espaço 3D não seja muito adequada para comparar duas cores [CJSW01].

Partindo da representação da cor através do modelo RGB , outros tipos de modelos de

cores podem ser derivados por transformações lineares ou não lineares do espaço RGB . Alguns modelos de cores obtidos por transformações lineares do modelo RGB herdam características como a alta correlação entre os seus canais e uma forte associação com a iluminação existente no ambiente. Por outro lado, os modelos obtidos por transformações lineares envolvem um menor esforço computacional para serem calculados que os de transformações não-lineares [CJSW01].

Os modelos YUV e YIQ são espaços de cores obtidos por transformações lineares e são utilizados para transmissão de TV (YUV no modelo de transmissão de vídeo PAL e o YIQ no modelo de transmissão de vídeo NTSC). Nesses padrões, o Y é o componente da luminância e o U , V , I e Q são os componentes relacionados à crominância (característica que define o tom e a quantidade de branco que uma cor contém).

Um outro modelo de cor obtido por transformação linear é o $I_1I_2I_3$. Esse modelo foi desenvolvido por Ohta [OKS80] depois de sucessivos testes com segmentação de imagens utilizando mais de 100 características de cores em oito tipos de imagens coloridas. Comparado com outros sete modelos de cores (RGB , YIQ , HSI , $Nrgb$, XYZ , $CIE(L * a * b^*)$ e $CIE(L * u * v^*)$) o modelo $I_1I_2I_3$ apresentou um melhor resultado em termos de qualidade da segmentação e da complexidade computacional exigida na transformação [OKS80]. O modelo pode ser obtido pelas equações

$$\begin{aligned} I_1 &= \frac{1}{3}(R + G + B), \\ I_2 &= \frac{1}{2}(R - B), \\ I_3 &= \frac{1}{4}(2G - R - B), \end{aligned} \tag{2.6}$$

onde o R , G e B são os valores dos canais do modelo RGB .

Já nos sistemas de cores obtidos por transformações não-lineares do espaço RGB , o problema da correlação entre os canais é minimizado solucionado [CJSW01]. Um dos modelos de cores dessa categoria é o RGB normalizado, denotado por $Nrgb$, onde a normalização nos canais RGB é feita para reduzir a influência da variação de intensidade na imagem, e assim aumentar a performance da segmentação. Segundo Terrillon [TDA98] a normalização reduz a sensibilidade da distribuição da cor tornando o modelo mais robusto à variação da iluminação, mas por outro lado, ela torna a imagem muito ruidosa se ela for adquirida sobre baixa iluminação. O modelo $Nrgb$ pode ser obtido pelas equações

$$\begin{aligned}
 r &= \frac{R}{R + G + B}, \\
 g &= \frac{G}{R + G + B}, \\
 b &= \frac{B}{R + G + B},
 \end{aligned}
 \tag{2.7}$$

onde o R , G e B são os valores dos canais do modelo RGB .

O modelo HSI (*Hue, Saturation, Intensity*) é obtido por uma transformação não-linear, sendo comumente utilizado em processamento de imagens por apresentar uma maior intuitividade com o sistema visual do que os outros sistemas de cores mencionados acima. Isto acontece porque a visão humana pode distinguir diferentes cores (*hue*) mais facilmente do que diferentes intensidades ou saturações [CL94]. Neste sistema, a informação sobre cor é representada nos canais H (que representa a matiz) e S (que mede a pureza da cor, ou seja, quanto de pigmento branco está misturado ao pigmento básico), enquanto o canal I contém informações sobre a quantidade de pigmento preto que está misturado ao pigmento básico presente na imagem. Uma das desvantagens do sistema HSI é que ele apresenta instabilidade em imagens com baixa saturação, pois o valor do canal H é indefinido quando a saturação é zero [CJSW01]. O modelo HSI pode ser obtido pelas equações:

$$\begin{aligned}
 H &= \arctang\left(\frac{\beta}{\alpha}\right), \\
 S &= \sqrt{\alpha^2 + \beta^2}, \\
 I &= (R + G + B)/3,
 \end{aligned}
 \tag{2.8}$$

com

$$\begin{aligned}
 \alpha &= R - \frac{1}{2}(G + B), \\
 \beta &= \frac{\sqrt{3}}{2}(G - B),
 \end{aligned}
 \tag{2.9}$$

onde o R , G e B são os valores dos canais do modelo RGB .

A CIE (*Commission International de l'Eclairage*) desenvolveu em 1931 um modelo de cor perceptualmente uniforme e independente de dispositivos [CIE31], onde qualquer cor do espectro visível pode ser especificada pela combinação das variáveis X , Y e Z . Os valores de X , Y e Z podem ser computados por uma transformação linear do modelo RGB conforme a Equação 2.10.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.116 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}
 \tag{2.10}$$

Os modelos de cores $CIE(L * a * b^*)$ e $CIE(L * u * v^*)$ são dois exemplos de modelos de cores da CIE que podem ser obtidos por transformações não-lineares a partir do X , Y e Z . Segundo Cheng [CJSW01] os modelos de cores do CIE podem controlar as informações de cor e intensidade de forma mais independente e simples que o modelo de cor RGB , além de permitir uma direta comparação de cores baseada na separação geométrica no espaço de cores. O modelo de cor $CIE(L * a * b^*)$ pode ser obtido por

$$\begin{aligned} L &= 116(Y/Y_0)^{\frac{1}{3}} - 16, \\ a &= 500[(X/X_0)^{\frac{1}{3}} - (Y/Y_0)^{\frac{1}{3}}], \\ b &= 200[(Y/Y_0)^{\frac{1}{3}} - (Z/Z_0)^{\frac{1}{3}}], \end{aligned} \quad (2.11)$$

onde $Y/Y_0 > 0.01$, $X/X_0 > 0.01$ e $Z/Z_0 > 0.01$ e (X_0, Y_0, Z_0) são os valores de XYZ de referência para a cor branca. Já o modelo de cor $CIE(L * u * v^*)$ pode ser obtido por

$$\begin{aligned} L &= 116(Y/Y_0)^{\frac{1}{3}} - 16, \\ u &= 13L(u' - u_0), \\ v &= 13L(v' - v_0), \end{aligned} \quad (2.12)$$

onde $Y/Y_0 > 0.01$, Y_0 é o valor de referência de Y para a cor branca e u_0 e v_0 são as referências para a cor branca de u' e v' respectivamente. Os valores de u' e v' podem ser obtidos respectivamente pelas Equações 2.13 e 2.14.

$$u' = \frac{4X}{X + 15Y + 3Z} \quad (2.13)$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \quad (2.14)$$

2.2.2 Seleção de Informação de Cores

Uma imagem colorida é geralmente digitalizada no espaço de cores RGB , mas pode ser transformada para outros espaços, como foi descrito anteriormente, através de transformações lineares e não lineares do modelo RGB . Todos estes modelos podem ser utilizados para segmentação de imagens, mas nenhum deles se sobressai mais que os outros modelos na segmentação de todos os tipos de imagens coloridas [CJSW01]. A tarefa de selecionar o melhor espaço de cor ainda é uma das principais dificuldades na segmentação de imagens [GH92].

A segmentação de uma mesma imagem em espaços de cores (RGB , YUV , HSI , dentre outros) pode gerar diferentes resultados. Então, é interessante analisar o comporta-

mento das informações de cada canal de cor no resultado da segmentação com o objetivo de melhorar a qualidade da segmentação.

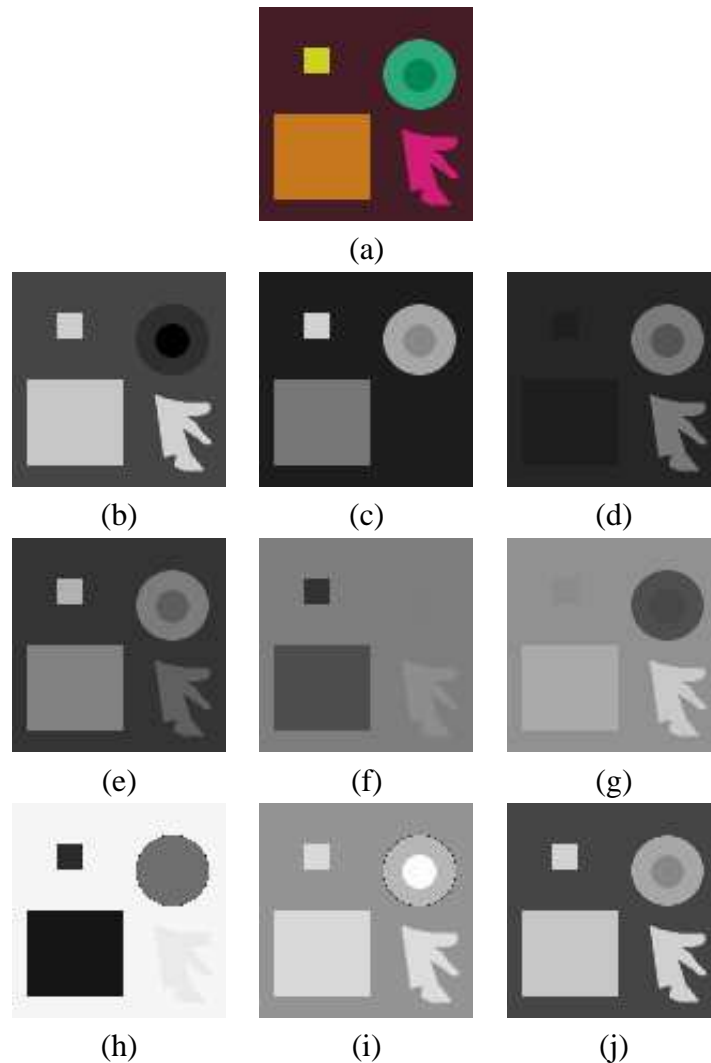


Figura 2.5: Análise isolada dos canais dos modelos RGB (b-d), YC_bC_r (e-g) e HSI (h-j) em uma imagem colorida (a).

Cada modelo de cor representa as informações de cores existentes na imagem de uma forma particular. Na Figura 2.5 o comportamento dos canais dos modelos RGB , YC_bC_r e HSI de uma imagem contendo seis objetos de diferente cores (marrom, laranja, amarelo, rosa, verde claro e verde escuro) pode ser analisado. A Figura 2.5 (a) apresenta uma imagem no modelo RGB , enquanto que cada canal separadamente (R, G e B) pode ser visto nas Figuras (b), (c) e (d) respectivamente. Pelo menos três regiões podem ser identificadas facilmente utilizando o canal B ou R. Na figura (e), (f) e (g) estão representados, respectivamente, os canais Y, C_b e C_r do modelo de cor YC_bC_r , sendo que pelo menos três regiões podem ser identificadas facilmente utilizando o canal C_b . Por fim, na figura (h), (i) e (j) estão representados respectivamente os canais H, S e I, sendo possível iden-

tificar facilmente pelo menos três regiões utilizando o canal H. Com a combinação dos canais B, C_b e H podem ser identificadas todas as regiões existentes na imagem.

Existem diversos trabalhos sobre o comportamento dos espaços de cores na segmentação de imagens. No trabalho desenvolvido por Cheng e outros [CJSW01] é apresentado um resumo das principais técnicas de segmentação de imagens coloridas, além de abordar o comportamento dos modelos de cores na segmentação desse tipo de imagem. No estudo desenvolvido por Gauch e Hsia [GH92] é feita uma comparação de três algoritmos de segmentação utilizando quatro espaços de cores, enquanto que Gevers e Smeulders [GS96] propõem um modelo de cor, que é independente da iluminação de uma cor e das discontinuidades provocadas pelas sombras, e baseado nas características de reflectância das superfícies dos objetos existentes na imagem. O trabalho apresentado por Liu e Yang [LY94] examina a utilização de seis diferentes espaços de cores na segmentação de imagens em multiresolução, e sua conclusão é que não existe um único espaço de cor que seja mais adequado para todas as imagens.

Um das estratégias para selecionar as melhores informações de cores para segmentação de imagem é a construção de modelos de cores híbridos. A idéia básica dos espaços de cores híbridos é combinar adaptativamente e/ou interativamente, diferentes canais de cores de vários espaços para: (a) aumentar a eficiência dos canais de cores para discriminação da informação de cor, e (b) reduzir a taxa de correlação entre os canais de cores [VMP03].

VandenBroucke [VMP98] propôs um método para classificar os pixels em um Espaço de Cores Híbrido (HSC- *Hybrid Color Space*) que é composto por um conjunto de três canais. Esse método apresenta um esquema de aprendizado supervisionado, baseado em um algoritmo *knock-out* [FHPZ96], que seleciona os melhores canais dentre um conjunto de modelos de cores pré-definidos para formar um espaço de cores híbridas que seja mais adequado para a segmentação. Ele utilizou um conjunto de 14 modelos de cores, formando um total de 42 canais de cores, de onde serão selecionados os três canais que melhor satisfaçam os critérios de compactação intra-objetos e separação inter-objetos proposto em seu trabalho. Como neste método a escolha dos canais é realizada através de treinamento, estes canais poderão produzir uma boa segmentação apenas para imagens semelhantes a que foram utilizadas para o treinamento.

Em uma outra metodologia, desenvolvida por Oliveira [OPC⁺06], a seleção das melhores informações de cores consiste em selecionar, dentre um conjunto de modelos de cores pré-definidos, os canais menos correlacionados entre si utilizando a correlação de Pearson. De acordo com Hair [Hai05], o coeficiente de correlação de Pearson mede a intensidade ou o grau de associação entre as variáveis analisadas, ou seja, mede a dependência linear entre duas variáveis. Essa metodologia seleciona três canais baseando-se

no princípio de que quanto mais diferentes forem os canais selecionados, mais informações estarão disponíveis para discriminar os objetos existentes na imagem. Os resultados dos experimentos mostraram que as segmentações de vídeos sintéticos e reais produzidas com a metodologia proposta obtiveram resultados semelhantes ou melhores que as obtidas usando os modelos de cores convencionais analisados [OPC⁺06].

2.2.3 Técnicas de Segmentação de Imagens Coloridas

A maioria das técnicas utilizadas para segmentação de imagens monocromáticas podem ser estendidas para imagens coloridas, como por exemplo, o *thresholding*, crescimento de regiões e a detecção de bordas. Nestas técnicas, a segmentação em imagens monocromáticas pode ser aplicada individualmente a cada canal da imagem colorida. O resultado da segmentação pode então ser produzido pela combinação dos canais utilizando algum método combinatório [CJSW01]. A segmentação de imagens coloridas é o processo de extração de uma ou mais regiões que satisfaçam um critério de homogeneidade pré-estabelecido, sendo esse critério baseado em informações de cores em um ou mais espaços de cores.

Uma questão interessante na adaptação das técnicas de segmentação monocromática para imagens coloridas é a de como iremos utilizar as informações de cor em um único pixel. No processo de digitalização da imagem, a cor que será atribuída a um pixel é decomposta nos três componentes de um modelo de representação de cor. Dessa forma, a informação é dividida e cada canal fica com uma parte da característica dessa cor; deste modo, a utilização direta dessas informações separadamente pode não ser a forma ideal para segmentar imagens coloridas.

A segmentação por *thresholding* para imagens coloridas, por exemplo, pode ser considerada uma extensão direta da segmentação de imagens monocromáticas. Nessa adaptação, é analisado isoladamente o histograma formado pelas informações existentes em cada canal do modelo de cores para gerar a segmentação. Uma limitação existente nessa adaptação é que quando o canal é analisado de forma independente para se determinar o limiar, a informação referente a cor original é perdida. Isso acontece porque o histograma de um canal é analisado isoladamente sem levar em consideração as informações de cores que os outros dois canais têm sobre a cor original do pixel. Por exemplo, um objeto em uma imagem pode ter o mesmo tom (*hue*) mas diferentes valores de saturação e luminosidade. Dependendo das combinações formadas pela divisão da informação de cor e do método utilizado para combinar os resultados da segmentação por *thresholding* em cada canal, o resultado final da segmentação pode não ser o desejado.

Neste trabalho segmentamos imagens coloridas utilizando informações de três canais

de qualquer modelo de cor. Primeiramente, o usuário selecionará pontos na região da imagem que ele deseja segmentar (pelo menos um ponto pra cada região). A partir das informações destes pontos, será definido um critério de similaridade que avaliará se um ponto pertence a uma região. Nesses critérios de similaridade, será atribuído um peso para cada canal analisado com o objetivo de privilegiar os canais que mais separam os objetos existentes na cena. As metodologias utilizadas para analisar e atribuir importância a cada canal serão discutidas na Seção 5.3.

Capítulo 3

Segmentação de Vídeos e Estimação de Movimento

3.1 Introdução

Avanços tecnológicos na produção de hardware e a padronização dos formatos de vídeos existentes têm contribuído para a viabilidade da transmissão, armazenamento, processamento e visualização em tempo real de vídeos digitais e para o seu compartilhamento entre diferentes áreas do conhecimento. Juntamente com essa evolução, cresce a necessidade da existência de técnicas e ferramentas para gerenciar e processar as informações existentes nos vídeos. A segmentação de vídeos é uma das primeiras etapas em muitas aplicações, que dentre outras coisas, analisam, codificam e transmitem vídeos digitais.

Um vídeo digital pode ser definido como uma seqüência de imagens bidimensionais que foram projetadas a partir de uma cena tridimensional [WOZ02]. Essa seqüência de imagens pode ser considerada como uma distribuição de intensidades no espaço (2D) variando no tempo. Uma representação para essa variação da imagem no tempo, pode ser expressa por $S_c(x_1, x_2, t)$, onde x_1 e x_2 são as coordenadas do espaço 2D e t é a variável referente ao tempo [Tek95].

Para Bovik [Bov05], a segmentação de vídeos digitais consiste na identificação de regiões que são homogêneas de acordo com algum critério pré-definido. Esses critérios têm o objetivo de analisar a similaridade entre os pixels dos *frames*, e assim maximizar a heterogeneidade dos pixels de diferentes regiões e a homogeneidade dos pixels pertencentes a uma mesma região. As características mais utilizadas para estabelecer esse critério de similaridade na segmentação de vídeos são a intensidade, a cor, a textura, o movimento e algumas propriedades geométricas. A utilização de diferentes características e critérios de homogeneidade para a segmentação de um mesmo vídeo pode gerar diferentes resultados.

A segmentação de um vídeo é diferente da segmentação isolada de cada *frame* desse vídeo. Isso acontece devido à necessidade de se observar uma consistência temporal entre os *frames* [Bov05]. Na segmentação de uma cena (*shot*) em um vídeo, é necessário que o resultado da segmentação gerado em um *frame* seja consistente com o resultado produzido nos *frames* adjacentes, enquanto que a segmentação de um único frame é um processo que não depende de informações dos *frames* adjacentes.

A segmentação de um mesmo vídeo poderá gerar diferentes resultados dependendo do nível de abstração desejado. Por exemplo, observando-se uma cena na natureza, uma floresta pode ser considerada como um único objeto, ou então, cada árvore pode ser considerada como diferentes objetos, ou ainda as características individuais das árvores (como folhas, galhos, troncos ou frutos) também podem ser considerados objetos distintos. Isso pode ser facilmente observado quando uma câmera faz um *zoom* em uma cena. Logo, a compreensão de que foi realizada uma segmentação apropriada da cena depende do nível de detalhamento que se deseja obter.

3.2 Técnicas para Segmentação de Vídeos

Muitas técnicas de segmentação de vídeos têm sido utilizadas em aplicações para o armazenamento, recuperação e reconhecimento de informações existentes em vídeos [Bov05], na criação de animações e renderizações não realists [CBNO06, WXSC04], em sistemas para compressão de vídeos [Che93] ou em em sistemas de monitoramento de tráfego de automóveis [VCPB03, Bev02]. Essas técnicas têm sido desenvolvidas com a intenção de superar as dificuldades inerentes à segmentação de vídeos, e atender as novas necessidades dos sistemas que utilizam a segmentação em uma de suas etapas de funcionamento.

Como um vídeo pode ser tratado como uma seqüência de imagens, a segmentação de vídeos herda algumas das dificuldades existentes na segmentação de imagens como o tratamento de imagens corrompidas por ruídos, iluminação não uniforme e background irregular. Além disso, a segmentação de vídeos tem suas dificuldades particulares, como a dificuldade de se processar vídeos contendo objetos não rígidos e movimentos bruscos.

Conforme Porikli e Wang [PW04], as técnicas de segmentação de vídeos podem ser agrupadas em três classes: métodos baseados em regiões, métodos baseados em *tracking* e métodos baseados em movimento. Os métodos de segmentação de vídeos baseados em regiões, utilizam critérios de homogeneidade para avaliar as características dos pixels de cada frame. A partir dessas informações, podem ser utilizados *clusters* ou algoritmos de divisão e união de regiões para segmentar os *frames*. As técnicas que utilizam *clusters* geralmente usam os histogramas (das intensidades ou canais) e os seus resultados são satisfatórios quando a distribuição de cor é multi-modal (*multi-modal color distribution*) ou

quando as informações de entrada são simples e com poucos ruídos [PW04]. A desvantagem da utilização do histograma é que ele falha em estabelecer conectividade espacial entre os pixels. As técnicas baseadas em divisão e união de regiões, geralmente apresentam melhor desempenho em termos de conectividade espacial e precisão nas bordas do que os métodos baseados em histogramas. Mas, as regiões detectadas podem não corresponder à forma do objeto se o background apresentar pixels com a mesma intensidade ou cor do objeto.

Uma outra classe de técnicas para segmentação de vídeos definida por Porikli e Wang, são os métodos baseados em *tracking*. Estes métodos têm o objetivo de encontrar a trajetória de um parâmetro dinâmico (como bordas, texturas ou regiões, dentre outras) que possua alguma conexão entre os *frames*. O *tracking* é muito utilizado em conjunto com outras técnicas de segmentação de vídeo. Como exemplo, existem algumas técnicas que propagam no tempo os objetos segmentados no primeiro frame de um vídeo, como é o caso da técnica de segmentação de vídeo apresentada por Grinias [GT01]. Uma das principais dificuldades na segmentação de vídeos baseada em *tracking* apontadas por Porikli [PW04], é o tratamento do movimento dos objetos entre os *frames*. Isso acontece devido à possibilidade da alteração da forma dos objetos durante o vídeo. Para solucionar esse problema, pode ser utilizado um modelo que especifique o comportamento do objeto no decorrer do tempo.

A última classe utiliza movimento como critério de similaridade. Em várias aplicações que utilizam a segmentação de vídeos, deseja-se identificar o objeto e principalmente encontrar a direção para onde o objeto está se deslocando. O objetivo dos métodos de segmentação baseados em movimento, é rotular os pixels ou objetos de cada frame que estão associados a movimentos existentes em uma cena. Esses métodos partem da hipótese de que os objetos presentes no vídeo têm movimentos coerentes (movimentos que possuam alguma ligação de um frame para o outro) e que possam ser modelados por um conjunto de parâmetros.

A segmentação de vídeos baseada na informação de movimento está fortemente relacionada a dois outros problemas do processamento de vídeos: a detecção e a estimação de movimento [Bov05]. A detecção é um tipo especial de segmentação baseada em movimento em que cada frame possui somente dois tipos de regiões: regiões estáticas e regiões que se movimentam (no caso de câmeras estáticas); e regiões com movimento local e movimento global (no caso de câmeras em movimento). A estimação de movimento, que será utilizada nesse trabalho, será descrita mais detalhadamente a seguir.

3.3 Estimação de Movimento

Devido à capacidade de armazenar informações não estáticas, o vídeo possui uma maior riqueza de informação que uma simples imagem. Enquanto uma imagem captura informações de uma cena em um instante no tempo, uma seqüência de imagens captura a dinâmica da cena. Dois motivos são destacados por Tekalp [Tek05] para ressaltar a importância da estimação de movimento no processamento de vídeos. O primeiro é o fato do movimento possuir informações importantes sobre a natureza dos objetos e a interação que esses objetos possuem em um vídeo. O outro motivo é que algumas características da imagem, como intensidade e cor, têm uma alta correlação com a direção do movimento, isto é, pode não existir uma troca significativa de valores de cor e intensidade de um objeto no decorrer do tempo.

Para que as informações sobre movimento possam ser estudadas, foram criados alguns modelos com o objetivo de representar formalmente os movimentos existentes em uma cena [Tek95]. Nesses modelos está representados tanto o movimento 3D do objeto na cena quanto o movimento 3D da câmera na cena. O modelo de coordenadas cartesianas, que é um exemplo desses modelos, representa movimento como a soma de rotações e translações 3D. Já no modelo homogêneo de coordenadas, o movimento é formado por transformações lineares do modelo cartesiano. Esses modelos são utilizados para representar o movimento 3D do conjunto de *pixels* que representa um objeto ou/e o movimento da câmera.

Nos modelos citados acima, é assumido que os objetos existentes na cena são objetos rígidos. Segundo a cinemática clássica, o movimento 3D pode ser classificado como movimento rígido ou não-rígido. No caso de movimento rígido, as distâncias relativas entre os conjuntos de pixels que formam os objetos permanecem fixas no decorrer do tempo, isto é, as estruturas 3D (forma) dos objetos em movimento podem ser modeladas como superfícies não deformáveis, como por exemplo, uma superfície plana ou superfície polinomial. Se o movimento for não-rígido, também conhecido com movimento deformável, será necessário a utilização de um modelo adicional para representar a estrutura do objeto durante a cena [Tek95]. No presente trabalho, no qual é utilizado o modelo de movimento rígido, a existência de movimento nos vídeos é resultado unicamente dos objetos, ou seja, nas seqüências utilizadas nos experimentos as câmeras são consideradas estáticas.

O objetivo da estimação do movimento 2D, também conhecido como velocidade instantânea, é encontrar o deslocamento dos pixels pertencentes a um objeto entre os instantes t e $t + 1$ (ver Figura 3.1). Em algumas situações, o deslocamento 2D do objeto pode ser diferente do valor estimado devido a dois fatores: variação insuficiente de intensidade (ou cor) em uma região do objeto que está se movendo (o centro de uma bola girando

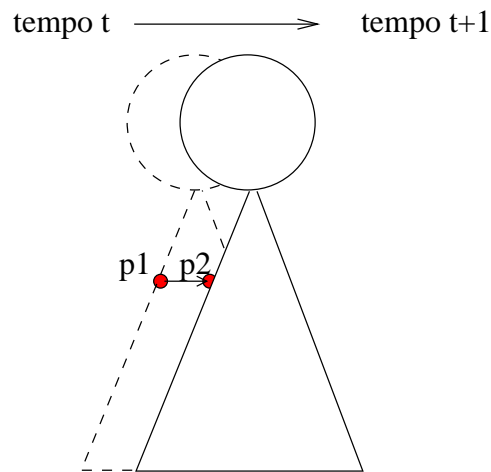


Figura 3.1: Estimação de movimento. O deslocamento do pixel p (de p_1 para p_2) entre os instantes t e $t+1$.

ou uma região chapada de um objeto em movimento) e a variação de iluminação em um objeto que está parado [VP89].

Existem vários métodos na literatura que propõem a estimação do movimento em vídeos digitais. Os métodos baseados na equação do fluxo óptico [HS81, LK81, Sin91] estimam o movimento em termos do gradiente espaço-temporal das intensidades na imagem. Já nos métodos baseados em movimento de blocos, é admitido que as imagens sejam formadas por blocos que podem se mover no decorrer do tempo. O cálculo da estimação do movimento para esse método pode ser feito utilizando técnicas como *Phase-Correlation* [JW87, ESE93] ou *Block-Matching* [Bie88b, GM90]. Nos métodos de estimação de movimento baseados em predição (*pel-recursive methods*) [Bie88a], as predições dos valores dos deslocamentos dos pixels podem ser encontradas através da estimação dos movimentos dos pixels no frame anterior ou por uma combinação linear da estimação do movimento na vizinhança dos pixels.

Dois fatores motivaram a utilização da estimação do movimento baseado na equação do fluxo óptico. O primeiro deles é a ampla utilização de métodos de estimação do movimento na segmentação de imagens e vídeos [LL06, HS99, GL00]. Uma outra motivação é o custo computacional razoável e a boa precisão que alguns algoritmos de estimação de movimento por cálculo de fluxo óptico tem apresentado [MNCG01, PGPO94].

3.4 Fluxo Óptico

Fluxo óptico é uma aproximação do movimento baseada na derivada local de uma dada seqüência de imagens, isto é, em 2D o fluxo óptico especifica quanto cada pixel de uma imagem se moveu entre dois *frames* adjacentes, enquanto que em 3D ele especifica quanto cada voxel (abreviação do inglês, *volumetric element*) se moveu entre volumes adjacentes [BT05]. Uma definição clássica para fluxo óptico é dada por Horn e Schunck [HS81]: "Fluxo óptico é a distribuição aparente de velocidade que um padrão de intensidade apresenta quando se move em uma imagem".

Além de ser utilizado para segmentação de imagens e vídeos, o fluxo óptico pode estimar propriedades e parâmetros de um sensor de movimento [LP80], calcular o foco de expansão e tempo de colisão [BA90], na compressão de vídeos [Che93], e até mesmo no cálculo do fluxo de sangue e do movimento dos músculos do coração [BF80]. Para a utilização de fluxo óptico em visão computacional é necessário fazer uma análise do conteúdo das imagens e do movimento existente nelas, para que assim se possa escolher o algoritmo mais adequado para se calcular o fluxo óptico [BB95].

A hipótese inicial para o cálculo do fluxo óptico é que a iluminação em uma região particular de uma imagem permaneça aproximadamente constante por um curto período de tempo [HS81]. Formalmente, se $I(x, y, t)$ é a função da intensidade em uma imagem, então

$$I(x, y, t) \approx I(x + \delta x, y + \delta y, t + \delta t), \quad (3.1)$$

onde δx e δy são os deslocamentos locais, nas coordenadas x e y , respectivamente, da região (x, y, t) na imagem, em um dado tempo δt . Utilizando a série de Taylor para expandir o lado esquerdo da Equação 3.1 temos

$$I(x, y, t) = I(x, y, t) + \nabla I \cdot \delta x + \nabla I \cdot \delta y + \delta t I_t + O^2, \quad (3.2)$$

onde $\nabla I = (I_x, I_y)$ e I_t são as derivadas de primeira ordem de $I(x, y, t)$, e O^2 é o termo de segunda ordem, que geralmente não é considerado devido ao seu valor desprezível. Efetuando-se a subtração de $I(x, y, t)$ nos dois lados da Equação 3.2, desprezando o termo O^2 e dividindo a Equação 3.2 por δt temos

$$\nabla I \cdot \bar{v} + I_t = 0, \quad (3.3)$$

onde $\nabla I = (I_x, I_y)$ é o gradiente da intensidade e $\bar{v} = (u, v)$ é a velocidade instantânea (fluxo óptico) de um pixel (x, y) no instante t . A Equação 3.3 é conhecida como a equação

de restrição do fluxo óptico (*optical flow constraint equation*) e define uma única restrição a velocidade (ver Figura 3.2). Nesta figura a velocidade normal v_{\perp} é definida como um vetor perpendicular à linha de restrição do fluxo óptico, e contém a maior velocidade na linha de restrição do fluxo óptico. Essa restrição não é suficiente para calcular os dois componentes da equação de restrição do fluxo óptico (Equação 3.3).

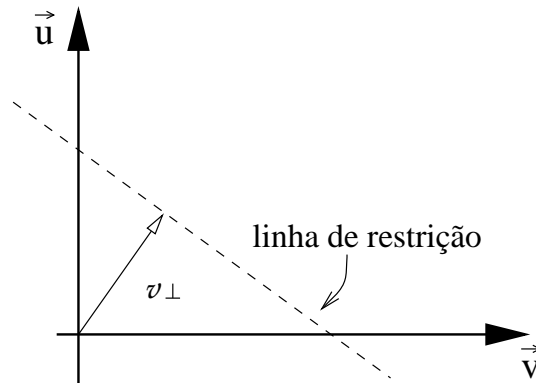


Figura 3.2: Equação de restrição do fluxo óptico. A reta produzida pela equação de restrição do fluxo óptico e a linha de restrição (pontilhada).

Dessa forma, somente o componente do movimento na direção do gradiente local da função de intensidade de uma imagem, a velocidade \bar{v}_{\perp} , pode ser estimada. Devido à velocidade normal estar na direção do gradiente espacial ∇I , a Equação 3.3 pode ser escrita como

$$v_{\perp} = \frac{-I_t \cdot \nabla I}{\|\nabla I\|_2^2}. \quad (3.4)$$

Assim, a medida da derivada espaço-temporal permite uma recuperação da velocidade normal. O problema do cálculo da velocidade completa da imagem se torna o de encontrar restrições adicionais para produzir uma segunda equação com as mesmas incógnitas [BT05].

Como já foi discutido anteriormente, o fluxo óptico calcula uma aproximação do movimento na imagem. Existem algumas condições que fazem a estimação do movimento, obtida através da utilização do fluxo óptico, seja diferente do movimento existente em uma seqüência de imagens. Dentre estas condições, podemos citar a ausência de textura, sombreamento, iluminação não-uniforme e superfícies oclusas. Um exemplo da implicação de uma dessas condições na estimação do movimento, é a existência de superfícies oclusas em um vídeo, que além de poder conter informações que aparecem e desaparecem, pode dificultar a análise da cena devido à possibilidade de haver informação de borda na superfície oclusa indicando a direção do deslocamento [BB95].

Existem diversos métodos para se calcular a estimação de movimento por fluxo óptico: métodos diferenciais, métodos baseados na frequência, métodos baseados na correlação, métodos de múltiplos movimentos e métodos por refinamento temporal. Em nosso trabalho, será utilizada uma técnica baseada nos métodos diferenciais, que podem calcular o movimento utilizando a derivada espaço-temporal das intensidades de uma imagem. Para isto é assumido que o domínio da imagem precisa ser contínuo (ou diferenciável) no espaço e no tempo.

Dentre as técnicas baseadas nos métodos diferenciais, as técnicas globais e locais utilizam as derivadas de primeira e de segunda ordem da Equação 3.3 para calcular o fluxo óptico. As técnicas globais adicionam uma restrição global na Equação 3.3, geralmente usam um termo para regular a suavização, para calcular o fluxo óptico entre dois *frames*. Já as técnicas locais usam o método dos mínimos quadrados para analisar a informação da velocidade normal na vizinhança e assim calcular o melhor valor do fluxo.

Para analisar alguns dos algoritmos que calculam o fluxo óptico, McCane e Galvin [MNCG01, GMN⁺98] apresentaram uma metodologia que avalia quantitativamente e qualitativamente algoritmos de fluxo óptico. Nessas investigações foram utilizadas seqüências de vídeos reais e sintéticas, e o grau de dificuldade de segmentação das seqüências foi baseado em dois princípios: complexidade da cena e movimentação dos objetos.

No trabalho desenvolvido por McCane [MNCG01], foram analisados sete algoritmos de fluxo óptico, sendo que o algoritmo de Proesmans [PGPO94] obteve um desempenho ligeiramente superior aos outros algoritmos em seqüência com vídeos reais. Em Galvin, no qual foram avaliados oito algoritmos, o de Proesmans foi o único algoritmo avaliado que produziu um vetor de fluxo consistente para cada pixel (ou seja, um mapa denso), apresentando sensível precisão, boa tolerância a ruídos e um moderado custo computacional [GMN⁺98].

O algoritmo de Proesmans é similar ao método de estimação de movimento apresentado por Horn e Schunck [HS81], sendo que a principal diferença é que ele incorpora um mecanismo de combinação na equação de restrição e um método pra tratar descontinuidade de fluxo que utiliza um processo conhecido como difusão anisotrópica para suavizar o fluxo. A implementação do algoritmo de Proesmans usada em McCane [MNCG01] é uma versão multi-resolução do algoritmo original. A aplicação da difusão anisotrópica significa que o somatório local dos componentes do fluxo óptico não é simplesmente a média, mas sim são atribuídos pesos de acordo com um mapa de consistência. Ao contrário do processo normal de suavização, o processo utilizado por McCane pode manter a descontinuidade em um fluxo e ainda manter um alto grau de continuidade em regiões locais [GMN⁺98].

Em nosso trabalho será utilizada uma versão multi-resolução do algoritmo de Proes-

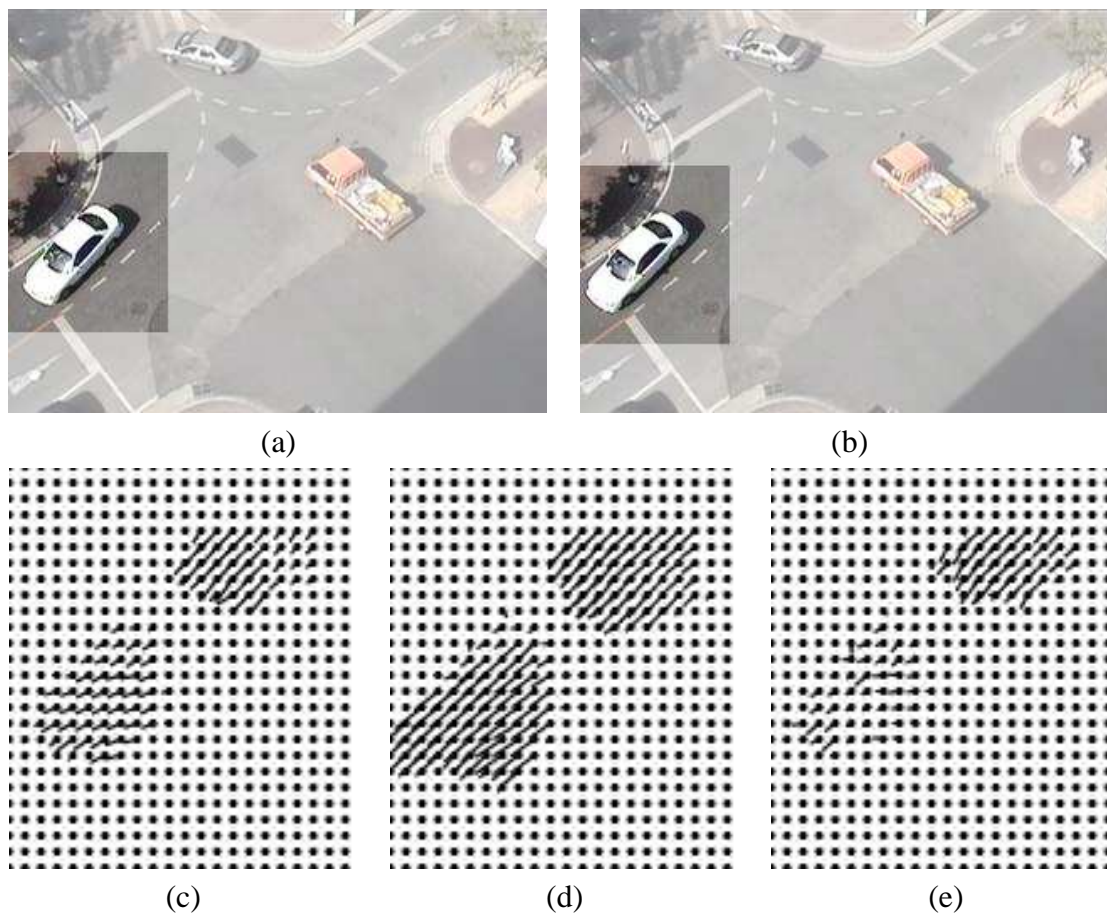


Figura 3.3: Estimação do movimento com diferentes parâmetros. Dois *frames* consecutivos de um vídeo (a) e (b) e a estimação de movimento na sub-imagem destacada em (a) e (b) utilizando o algoritmo de Proesmans com os seguintes parâmetros: (c) lambda 10, 5 níveis e 15 iterações; (d) lambda 1000, 5 níveis e 15 iterações; (e) lambda 1000, 2 níveis e 15 iterações.

desenvolvido por McCane [MNCG01]. Para calcular a estimativa de movimento utilizando esse algoritmo é necessário informar o número de iterações, o nível multi-resolução e o lambda (λ). A Figura 3.3 mostra a estimativa do movimento entre dois *frames* consecutivos de uma seqüência com diferentes valores para estes parâmetros. O primeiro parâmetro, número de iterações, é utilizado para indicar quantas vezes o fluxo óptico será calculado e refinado. O algoritmo de Proesmans utiliza um esquema dual para o refinamento do fluxo óptico, no qual é calculado o fluxo no sentido direto, do quadro n para o quadro $n + 1$, e no sentido inverso, do quadro $n + 1$ para o quadro n , o número de iterações indicará a quantidade de vezes que esse cálculo dos dois mapas será feito. O segundo parâmetro, o nível, indica em quantos níveis será analisado o fluxo óptico. Por exemplo, uma imagem com 256×256 pixels analisada em quatro níveis, será processada nas dimensões 256×256 , 128×128 , 64×64 e 32×32 . O terceiro, o λ , parâmetro indica a suavização que será aplicada no cálculo do fluxo óptico e influencia a difusão

anisotrópica.

Na Figura 3.3 (c), onde foram utilizados λ igual a 10, 5 níveis e 15 iterações, a diferença entre a intensidade do fluxo óptico entre os pixels do carro e a do asfalto é pequena. Isso ocorre devido ao baixo valor do λ utilizado na estimação do movimento. Já na Figura 3.3 (d), o aumento do λ para 1000 realça mais a diferença entre o fluxo dos objetos carro e asfalto. Figura 3.3 (e) mostra a influência do valor da quantidade de níveis analisados pelo algoritmo, onde a utilização de apenas dois níveis provoca imprecisão na intensidade e direção do fluxo.

Os valores dos parâmetros do algoritmo de Proesmans que produzem uma melhor estimação do movimento podem variar de acordo com o vídeo utilizado. A Figura 3.4 mostra o mapa do fluxo óptico calculado através do algoritmo do Proesmans para a seqüência *Demost*. Nessa seqüência existe uma boa diferença entre os vetores do fluxo óptico dos objetos em movimento e o background.

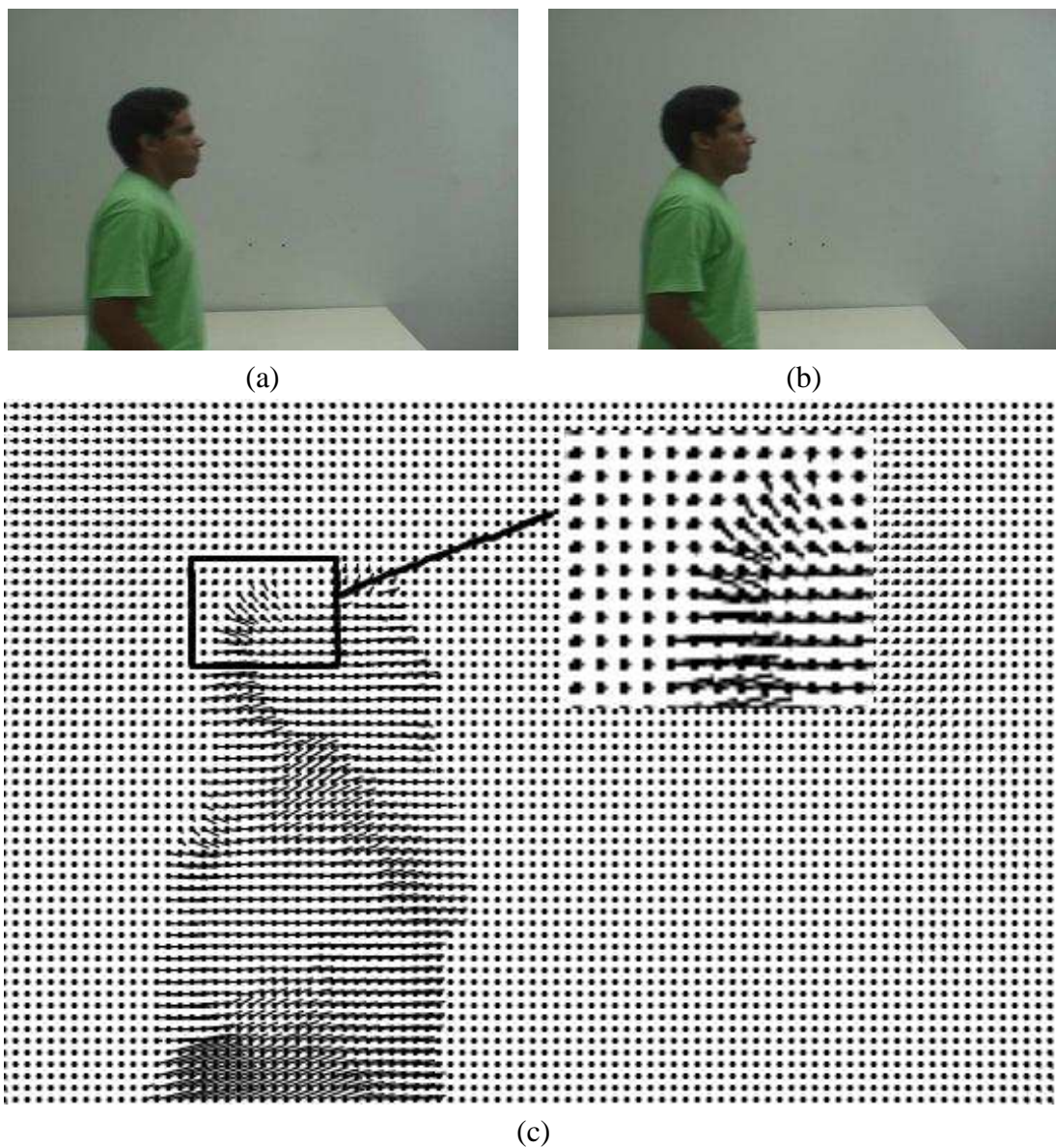


Figura 3.4: Exemplo da estimação de movimento. Dois *frames* de uma seqüência (a) (b) e o mapa do fluxo óptico (c) calculado através de uma versão multi-resolução do algoritmo de Proesmans [PGPO94]. Foram utilizados como parâmetros λ igual a 1000, 15 iterações e 4 níveis.

Capítulo 4

Segmentação Fuzzy de Imagens

4.1 Introdução

O processo de segmentar um objeto em um background de uma imagem é uma tarefa difícil para um computador quando a característica que difere entre o objeto e o background é alguma propriedade textural em vez da intensidade, ou então, a imagem está corrompida por ruído ou por uma iluminação não homogênea [CGHK99]. A existência dessas interferências na imagem exige dos algoritmos de segmentação um esforço adicional para identificar e separar os objetos presentes nas imagens.

Em uma segmentação que utiliza métodos baseados em *thresholding*, cada pixel da imagem é comparado com um pré-definido limiar t para verificar se ele pertence ou não ao objeto. Em algumas imagens, devido à distribuição dos valores da intensidade nos pixels, não é possível encontrar um limiar t que seja capaz de separar os objetos em uma imagem. A Figura 4.1 (a), por exemplo, possui um hexágono com um retângulo na sua parte superior que tem a sua intensidade decrescendo da direita pra esquerda. Conforme mostrado em Carvalho [CGHK99], é impossível encontrar um limiar t que possa segmentar o retângulo sem segmentar também parte do fundo da imagem, produzindo um resultado muito melhor que o resultado da Figura 4.1 (b). Uma das dificuldades da segmentação de imagens por *thresholding* é a presença de ruídos, a iluminação não uniforme e o background irregular.

A segmentação *fuzzy* é uma técnica que determina para cada elemento da imagem um grau de pertinência (entre zero e um) indicando a certeza desse elemento pertencer ou não a um determinado objeto existente na imagem [CHK05]. A partir dessas informações de pertinência, o algoritmo de segmentação pode tomar uma decisão mais flexível sobre a classificação de cada pixel.

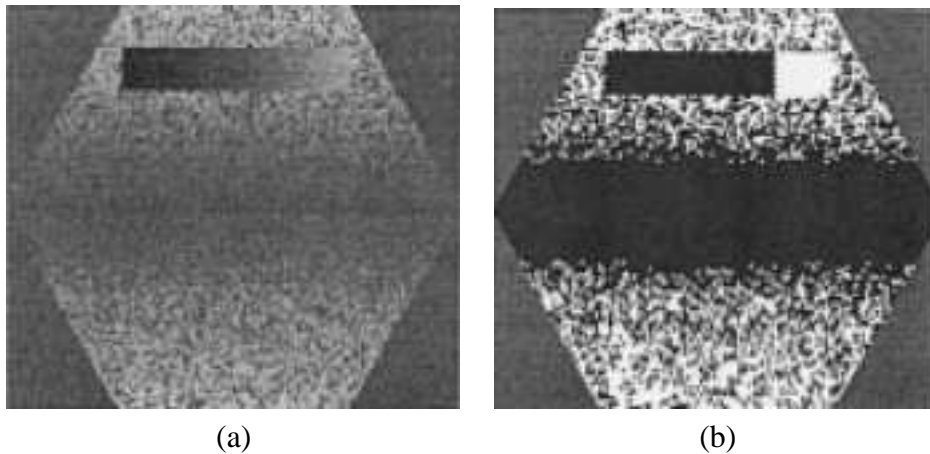


Figura 4.1: Exemplo da segmentação por *Thresholding*. Uma figura com seu real valor de intensidade (a) e uma imagem binária produzida por um limiar (b) (Figura extraída de Carvalho [CGHK99]).

O algoritmo de segmentação *fuzzy* (*MOFS - Mult-Object Fuzzy Segmentation*) desenvolvido por Herman e Carvalho [HC01] utiliza o conceito de *fuzzy connectedness* (conectividade *fuzzy*) proposto por Rosenfeld [Ros79] e foi inspirado no trabalho desenvolvido por Udupa e Samarasekera [US96] sendo generalizado para espaços digitais arbitrários conforme a definição de Herman [Her98].

4.2 Segmentação Fuzzy Simultânea de Múltiplos objetos

O algoritmo proposto por Herman e Carvalho [HC01] é um método semi-automático de segmentação por crescimento de regiões onde o usuário escolhe um ou mais pixels sementes para representar os objetos que se deseja segmentar. Como será explicado posteriormente o algoritmo MOFS não satisfaz a Condição ϵ da definição de segmentação baseada em regiões apresentada na Seção 2.1.1. Por ser um método interativo, seu sucesso está fortemente relacionado com a escolha dos pixels escolhidos como sementes de cada objeto. Este método não é específico para apenas uma área de atuação e pode ser utilizado em aplicações de clusterização [JMF99] e na segmentação de imagens, de um modo geral [COG06, CBNO06, HC01].

No trabalho de Herman e Carvalho [HC01], foi feita uma generalização do conceito de conectividade *fuzzy*, onde os elementos de um conjunto V são chamados de *spels* (do inglês *spatial elements*) e podem estar se referindo, dentre outras coisas, a *pixels* em uma imagem, pontos em um plano ou *voxels* em um volume. Nesse enfoque, para cada par de *spels* (c,d) é determinado um número real entre 0 e 1, que é referido como uma

conectividade *fuzzy* entre c e d . Essa abordagem gera um conceito de conjunto *fuzzy* como foi definido em Pal e Dutta-Majumber [PDM86], onde o conjunto *fuzzy* em questão é o conjunto de pares conectados e o grau de afinidade entre o par (c,d) é a conectividade *fuzzy*.

No decorrer da execução do algoritmo, os elementos do conjunto V são associados a um objeto utilizando o conceito *fuzzy*, isto é, para cada *spel* determina-se qual é o grau de pertinência que ele tem com o objeto ao qual foi associado (é atribuído um número entre 0 e 1, onde 0 indica que o *spel* definitivamente não pertence a um objeto e 1 que o *spel* definitivamente pertence ao objeto). Para fazer a análise de cada *spel* e encontrar o grau de pertinência que esse *spel* tem com cada objeto é utilizada uma função de afinidade *fuzzy*.

Baseado nessa função de afinidade foi criado o conceito de *M-semisegmentation* (*M*-semisegmentação). Uma *M-semisegmentation* de V é uma função σ que mapeia cada $c \in V$ em um vetor de $(M+1)$ -dimensões $\sigma^c = (\sigma_0^c, \sigma_1^c, \dots, \sigma_M^c)$, no qual

1. $\sigma_0^c \in [0, 1]$ (isto é, σ_0^c está entre 0 e 1),
2. para cada m ($1 \leq m \leq M$), o valor de σ_m^c é zero ou σ_0^c ,
3. pelo menos para um m ($1 \leq m \leq M$), $\sigma_m^c = \sigma_0^c$,

onde σ_m^c informa o grau de pertinência do *spel* c pertencer ao m -ésimo objeto, e σ_0^c é igual a $\max_{1 \leq m \leq M} \sigma_m^c$. Uma *M-semisegmentação* σ é uma *M-segmentação* se para todo *spel* $c \in V$, σ_0^c é positivo.

De acordo com a definição utilizada em Carvalho [CHK05] uma corrente (*chain*) é uma seqüência de distintos *spels* $\langle c^{(0)}, \dots, c^{(K)} \rangle$ e um elo (*link*) é um par ordenado $(c^{(k-1)}, c^{(k)})$ de consecutivos *spels* em uma corrente. A força desse elo também é um conceito *fuzzy*, ou seja, para cada par ordenado (c, d) de *spels* é atribuído um número real entre 0 e 1 que define a força do elo entre c e d . A ψ -força de um elo é um valor fornecido por uma função de afinidade *fuzzy* $\psi : V^2 \rightarrow [0, 1]$, isto é, uma função que atribui um número real positivo entre 0 e 1 para cada par ordenado de *spel*. A Figura 4.2 mostra duas correntes, uma azul e uma verde, pertencentes a dois objetos que estão sendo segmentados e pelo *spel* c' passam as duas correntes. Para a segmentação de múltiplos objetos é interessante que se defina uma função de afinidade para cada objeto. Isso se deve ao fato que geralmente cada objeto possui características particulares e a sua função de afinidade deve ser capaz de identificar essas características nos pixels das imagens.

A Condição e da Seção 2.1.1, que apresenta uma definição clássica de segmentação baseada em regiões, afirma que dado um predicado P e duas regiões adjacentes R_i e R_j ,

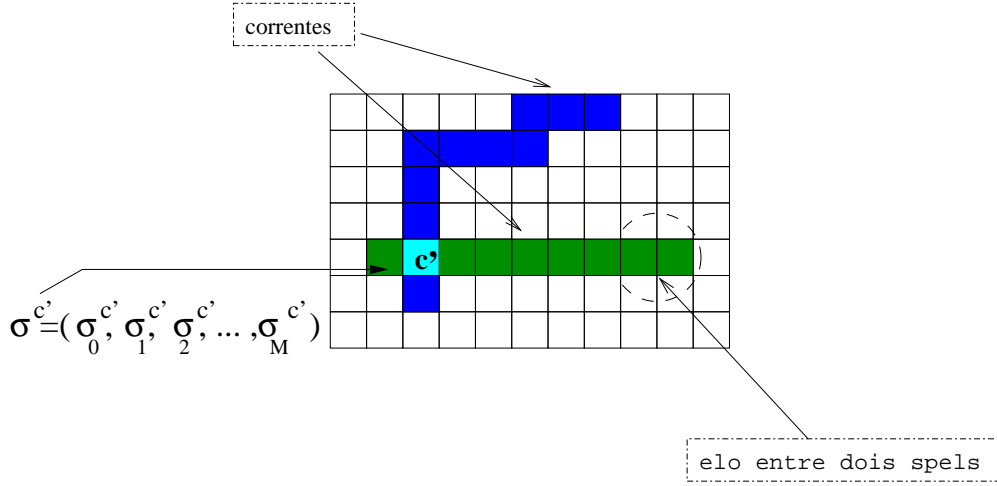


Figura 4.2: Exemplo de duas correntes existentes em uma imagem

$P(R_i \cup R_j) = \text{FALSE}$. Esta condição afirma que não existem duas regiões adjacentes que satisfaçam simultaneamente uma mesma condição. Já na segmentação *fuzzy* descrita aqui, um *spel* c pode ter uma mesma afinidade com duas correntes de dois objetos diferentes, ou seja, podem existir *spels* que pertençam a dois objetos ($\sigma^c = \sigma^c = \sigma_0^c$), negando assim a condição apresentada. As demais condições apresentadas na Seção 2.1.1 são satisfeitas pela segmentação *fuzzy*.

A ψ -força de uma corrente é a menor ψ -força de um elo pertencente a essa corrente e por definição a ψ -força de uma corrente que possui somente um *spel* é igual a 1. Dado um conjunto $U (\subseteq V)$ ele é dito ser ψ -conectado se para todos os pares de *spel* em U existe uma corrente em U de ψ -força positiva do primeiro *spel* para o segundo. Um grafo M -fuzzy é um par (V, Ψ) , onde V é um conjunto finito e $\Psi = (\psi_1, \dots, \psi_M)$ onde ψ_m (para $1 \leq m \leq M$) é uma função de afinidade *fuzzy*.

Um grafo M -fuzzy *semeado* (*seeded M-fuzzy graph*) é uma tripla (V, Ψ, \mathcal{V}) onde (V, Ψ) é um grafo M -fuzzy e $\mathcal{V} = (V_1, \dots, V_M)$, onde $V_m \subseteq V$ para $1 \leq m \leq M$ é o conjunto de todos os *spels* sementes. Um grafo M -fuzzy *semeado* $(V, (\psi_1, \dots, \psi_M), (V_1, \dots, V_M))$ é considerado *conectado* se

1. o conjunto V é ϕ_Ψ -conectado, onde $\phi_\Psi(c, d) = \min_{1 \leq m \leq M} \psi_m(c, d)$ para todo $c, d \in V$, e
2. $V_m \neq \emptyset$, para pelo menos um $m, 1 \leq m \leq M$.

Para uma M -semisegmentação σ de V e $1 \leq m \leq M$, uma corrente $\langle c^{(0)}, \dots, c^{(K)} \rangle$ é definida como uma σm -corrente se $\sigma_m^{c^{(k)}} > 0$, para $0 \leq k \leq K$. Além disso, para

todo $W \subseteq V$ e $c \in V$, será usado $\mu_{\sigma,m,W}(c)$ para denotar a ψ_m -força máxima de uma σm -corrente de um *spel* em W para c (essa força será 0 se tal corrente não existir).

Teorema 1 *Se (V, Ψ, \mathcal{V}) é um grafo M -fuzzy semeado (onde $\Psi = (\psi_1, \dots, \psi_M)$ e $\mathcal{V} = (V_1, \dots, V_M)$), então*

1. *existe uma M -semisegmentação σ de V com as seguintes propriedades : para todo $c \in V$, se para $1 \leq n \leq M$*

$$s_n^c = \begin{cases} 1, & \text{se } c \in V_n, \\ \max_{d \in V} (\min(\mu_{\sigma,n,V_n}(d), \psi_n(d, c))), & \text{caso contrário,} \end{cases} \quad (4.1)$$

então para $1 \leq m \leq M$

$$\sigma_m^c = \begin{cases} s_m^c, & \text{if } s_m^c \geq s_n^c, \text{ for } 1 \leq n \leq M, \\ 0, & \text{caso contrário,} \end{cases} \quad (4.2)$$

2. *esta M -semisegmentação é única; e*
3. *ela é uma M -segmentação, desde que (V, Ψ, \mathcal{V}) seja conectado.*

A prova desse teorema foi publicada por Carvalho et al. [CHK05]. Uma descrição gráfica desse teorema pode ser vista na Figura 4.3. Suponha que na Figura 4.3 c seja um *spel* arbitrário e que σ^d seja conhecido para todos os outros *spels* d . Então, para $1 \leq n \leq M$ ($M=3$ na Figura 4.3), o s_n^c da Equação 4.1 é a ψ_n -força máxima de uma corrente $\langle d^{(0)}, \dots, d^{(L)}, c \rangle$ de um *spel* semente em V_n para c de modo que $\sigma_n^{d^{(l)}} > 0$ (isto é, $d^{(l)}$ pertença ao n -ésimo objeto) para $0 \leq l \leq L$. Intuitivamente, o n -ésimo objeto pode afirmar que c pertença a ele se, e somente se, s_n^c é máximo. Isto confirma o que é proposto pela Condição 4.2: σ_m^c tem um valor positivo somente para o objeto cujo valor de s_m^c é o máximo. Além disso, esta propriedade nos mostra como um *spel* pode se relacionar com vários objetos, desde que σ seja conhecida para todos os outros *spels*, e com a Condição 4.2 satisfeita, pode-se calcular os valores de s_m^c para um *spel* c utilizando-se a Equação 4.1. O Teorema 1 diz que existe uma, e somente uma, M -semisegmentação que satisfaz simultaneamente as proposições, e que esta M -semisegmentação é de fato uma M -segmentação determinada pelo fato do grafo M -fuzzy semeado ser conectado.

4.3 Algoritmo para Segmentação Fuzzy

Em Herman e Carvalho [HC01] foi apresentado um algoritmo guloso eficiente, o MOFS (Multi-Objetct Fuzzy Segmentation), cujo resultado é uma M -segmentação σ de acordo

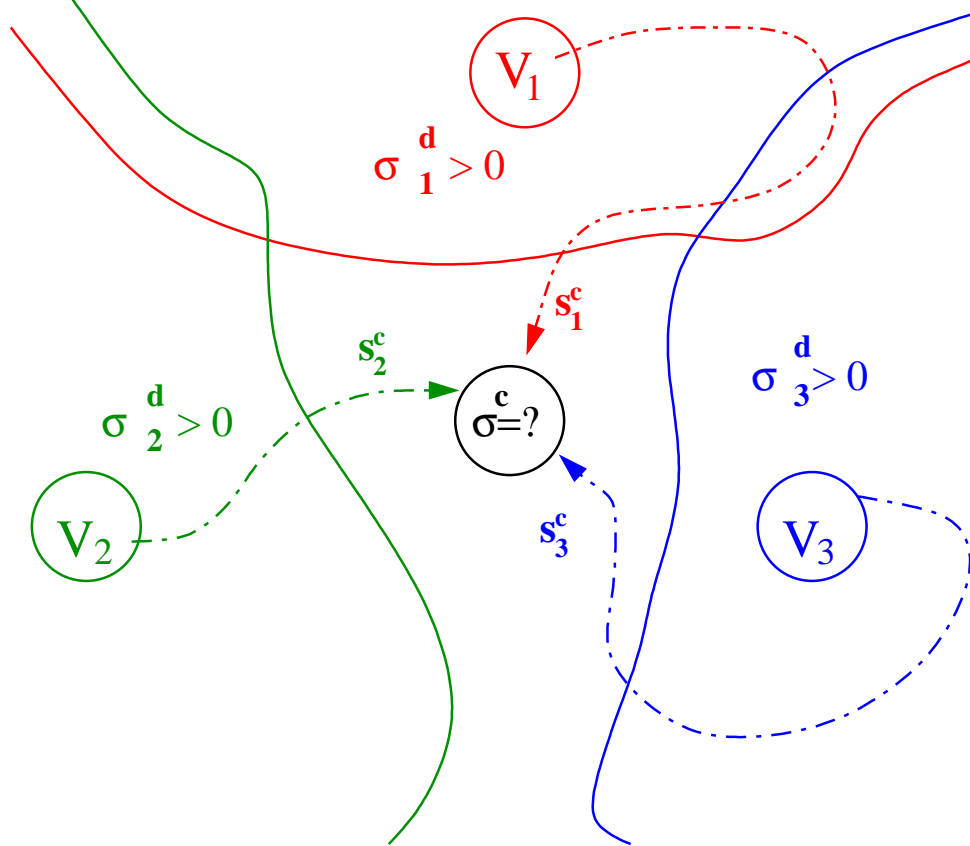


Figura 4.3: Ilustração gráfica do Teorema 1.

com o Teorema 1. Uma implementação direta desse algoritmo pode não ser computacionalmente eficiente devido à possibilidade da criação de passos que não produzam troca de status, ou seja, passos nos quais não ocorrem alterações na distribuição dos *spels* entre as regiões. Esses passos foram evitados com uma implementação mais cuidadosa do algoritmo, evitando assim desperdício de recursos e ainda produzindo os mesmos resultados [CHK05].

Esse algoritmo utiliza um heap binário H de *spels* c , com uma chave associada σ_0^c [CLR90]. O valor da chave do *spel* que está no topo do heap é o valor máximo do heap, e esse valor é definido por Chave-máxima (H) (este valor é zero se o heap estiver vazio). Na execução do algoritmo, cada *spel* é inserido em H exatamente uma única vez (utilizando a operação $H \leftarrow H \cup \{c\}$), e eventualmente removido de H com a operação $\text{Remove-Max}(H)$ onde é removido o *spel* que está no topo do heap. No momento que o *spel* c é removido de H , o σ^c tem o seu valor final. Os *spels* são removidos do heap H em uma ordem decrescente do valor final σ_0^c e r é utilizado para armazenar o atual valor da Chave-máxima (H).

1- Algoritmo MOFS

1. **for** $c \in V$ **do**
2. **for** $m \leftarrow 0$ **to** M **do**
3. $\sigma_m^c \leftarrow 0$
4. $H \leftarrow \emptyset$
5. **for** $m \leftarrow 1$ **to** M **do**
6. $U_m \leftarrow V_m$
7. **for** $c \in U_m$ **do**
8. **if** $\sigma_0^c = 0$ **then do** $H \leftarrow H \cup \{c\}$
9. $\sigma_0^c \leftarrow \sigma_m^c \leftarrow 1$
10. $r \leftarrow 1$
11. **while** $r > 0$ **do**
12. **for** $m \leftarrow 1$ **to** M **do**
13. **while** $U_m \neq \emptyset$ **do**
14. remova um spel d de U_m
15. $C \leftarrow \{c \in V \mid \sigma_m^c < \min(r, \psi_m(d, c)) \text{ and } \sigma_0^c \leq \min(r, \psi_m(d, c))\}$
16. **while** $C \neq \emptyset$ **do**
17. remova um spel c de C
18. $t \leftarrow \min(r, \psi_m(d, c))$
19. **if** $r = t$ **and** $\sigma_m^c < r$ **then do** $U_m \leftarrow U_m \cup \{c\}$
20. **if** $\sigma_0^c < t$ **then do**
21. **if** $\sigma_0^c = 0$ **then do** $H \leftarrow H \cup \{c\}$
22. **for** $n \leftarrow 1$ **to** M **do**
23. $\sigma_n^c \leftarrow 0$
24. $\sigma_0^c \leftarrow \sigma_m^c \leftarrow t$
25. **while** Chave-máxima (H) = r **do**
26. Remove-Max(H)
27. $r \leftarrow$ Chave-máxima (H)
28. **for** $m \leftarrow 1$ **to** M **do**
29. $U_m \leftarrow \{c \in H \mid \sigma_m^c = r\}$

O processo é iniciado (passos 2-9) primeiramente colocando zero em σ_m^c para todos os *spels* $c \in V$ e $0 \leq m \leq M$. Então, para todos os *spels* semente $c \in V_m$, c é adicionado em U_m e em H e os valores de σ_0^c e σ_m^c atualizados para 1, indicando que os mesmos pertencem aos respectivos objetos. No final da inicialização do algoritmo as seguintes condições são satisfeitas:

- (i) σ é uma M -semisegmentação de V .
- (ii) Um *spel* c está em H se, e somente se, $0 < \sigma_0^c \leq r$.
- (iii) $r = \text{Chave-máxima}(H)$.
- (iv) Para $1 \leq m \leq M$, $U_m = \{c \in H \mid \sigma_m^c = r\}$.

A inicialização vem seguida pelo laço principal do algoritmo. No início da execução de cada laço, as condições de (i) a (iv) são satisfeitas. O laço principal é repetido até o valor de r ser igual a 0 (quando o algoritmo termina no Passo 29) e é dividido em duas partes, cada uma com uma funcionalidade diferente.

A primeira parte do laço principal (Passos 10-24) é responsável pela atualização da suposição atual para o valor final de σ_m^c . O valor de σ_m^c é atualizado para um valor maior se é encontrado uma σm -corrente de um *spel* semente em V_m para c com ψ_m -força maior que o antigo valor, e ele é atualizado para zero se ele encontra (para $n \neq m$) uma σ_m -corrente de um *spel* semente em V_m para c de ψ_m -força maior que o antigo valor de σ_m^c . Na segunda parte do laço principal (Passos 25-29), as condições (iii) e (iv) são restauradas para o novo valor de r . Nessa parte são retirados do *heap* todos os *spels* que tiverem uma afinidade igual a r . Logo após essas remoções, o valor do r é atualizado para o valor máximo do *heap*, e nos passos 28 e 29 os *spels* que tiverem afinidades iguais a r serão armazenados em U_m .

4.4 Algoritmo Rápido para Segmentação Fuzzy

Embora o algoritmo apresentado anteriormente seja capaz de segmentar imagens 3D com mais de 7 milhões de *spels* em aproximadamente 4 minutos, este tempo de resposta pode não ser adequado para alguns tipo de aplicações que necessitem de maior rapidez no processo de segmentação [CHK05]. Em Carvalho [CHK05] foi apresentado um algoritmo rápido de segmentação *fuzzy* que pode ser aplicado em algumas circunstâncias nas quais a qualidade da segmentação não seja consideravelmente afetada por uma discretização dos valores gerados pela função de afinidade.

Supondo que um conjunto não vazio R de afinidades fuzzy, que atribui valores para *spels* de uma classe particular de problema, seja sempre um subconjunto do conjunto A e K seja a cardinalidade do conjunto $A \cup \{1\}$ com $1 = a_1 > a_2 > \dots > a_K > 0$ sendo os elementos de A . As funções de afinidade *fuzzy* podem ser definidas assim pois em muitas aplicações, a qualidade de segmentação fuzzy não é significativamente afetada se cada valor da afinidade for arredondado para, por exemplo, três casas decimais. Se for utilizado o arredondamento para as afinidades dos *spels*, poderá ser utilizado um

conjunto $A = \{0.001, 0.002, \dots, 0.999, 1.000\}$, e então $K = 1000$ e $a_k = 1.001 - k/1000$ [CHK05]. Isso permite que usemos listas para armazenar os valores de σ_0^c ao invés de um *heap* binário.

2 - Algoritmo MOFS-*fast*. Versão rápida do algoritmo MOFS

1. **for** $c \in V$ **do**
2. **for** $m \leftarrow 0$ **to** M **do**
3. $\sigma_m^c \leftarrow 0$
4. **for** $m \leftarrow 1$ **to** M **do**
5. **for** $c \in V_m$ **do**
6. $\sigma_0^c \leftarrow \sigma_m^c \leftarrow 1$
7. $U[m][1] \leftarrow V_m$
8. **for** $k \leftarrow 2$ **to** K **do**
9. $U[m][k] \leftarrow \emptyset$
10. **for** $k \leftarrow 1$ **to** K **do**
11. **for** $m \leftarrow 1$ **to** M **do**
12. **while** $U[m][k] \neq \emptyset$ **do**
13. remova um spel d do conjunto $U[m][k]$
14. $C \leftarrow \{c \in V \mid \sigma_m^c < \min(a_k, \psi_m(d, c)) \text{ and } \sigma_0^c \leq \min(a_k, \psi_m(d, c))\}$
15. **while** $C \neq \emptyset$ **do**
16. remova um spel c de C
17. $t \leftarrow \min(a_k, \psi_m(d, c))$
18. **if** $\sigma_0^c < t$ **then do**
19. remova cada c do conjunto U que satisfaça a condição
20. **for** $n \leftarrow 1$ **to** M **do**
21. $\sigma_n^c \leftarrow 0$
22. $\sigma_0^c \leftarrow \sigma_m^c \leftarrow t$
23. insira c no conjunto $U[m][l]$ onde l é o inteiro que $t a_l = t$

Nesta nova implementação, que foi utilizada nos experimentos deste trabalho, ao invés de utilizar uma fila de prioridade H (como foi utilizada na implementação do algoritmo) foi utilizada um matriz de $M \times K$ denominada $U[m][k]$, que armazena os conjuntos de

nós que representam cada *spel* (M representa o número de objetos e K a cardinalidade do conjunto A). Esta implementação é mais eficaz (com uma complexidade de espaço na ordem de $O(M(K + V))$) se todas as estruturas de dados puderem ser armazenadas na memória principal.

Na primeira implementação do algoritmo MOFS, o custo de tempo para atualizar a *heap* H era da ordem de $O(\log N)$ para cada aumento da força σ_0^c de um *spels* c , onde N é o número de *spel* no *heap* em um instante t . Na nova implementação, quando o valor de qualquer componente σ_0^c é trocado, é feita a alteração no conjunto de nós correspondentes, com o custo de tempo na ordem de $O(\max(1, |\{m \geq 1 | \sigma_0^c > 0\}|))$.

A Figura 4.4 (a) e (b) apresenta duas fatias de uma Tomografia Computadorizada (CT) e as suas correspondentes 4-segmentações utilizando o algoritmo original (Figura 4.4 (c) e (d)) e a nova implementação do MOFS (Figura 4.4 (e) e (f)). A nova implementação precisa de aproximadamente 35s para produzir uma 4-segmentação de 164 frames (aproximadamente $5\mu s$ por *spel*) comparados com os 249s de execução do algoritmo original (aproximadamente $34\mu s$ por *spel*).

4.5 Exemplos de Segmentação Fuzzy de Imagens

Existem diversas possibilidades de aplicações da segmentação *fuzzy* conforme citado anteriormente na Seção 4.2. Aqui mostraremos com mais detalhes dois exemplos de aplicação da segmentação *fuzzy* em imagens. O primeiro mostra uma segmentação de imagens de ressonância magnética enquanto que o segundo mostra uma seleção semi-automática de partículas em micrografos.

Para a segmentação das imagens, é necessário definir ψ_m e V_m com $1 \leq m \leq M$, ou seja, as funções de afinidade e os conjuntos dos pixels sementes. O conjunto V_m é formado por alguns pixels e os seus vizinhos de borda, da imagem que pertencem ao m -ésimo objeto. Neste caso, para a definição de ψ_m com $1 \leq m \leq M$, é assumido que importantes características de um objeto são definidas pelos valores dos pixels da imagem em uma região, e pela diferença desses valores na sua vizinhança [CGHK99]. Deste modo, para estes dois exemplos foram utilizadas funções de afinidades da forma:

$$\psi_m(c, d) = \begin{cases} 0, & \text{se } c, d \text{ não são adjacentes,} \\ [\rho_{g_m, h_m}(g) + \rho_{a_m, b_m}(a)] / 2, & \text{caso contrário,} \end{cases} \quad (4.3)$$

onde g_m é a média e h_m é o desvio padrão para a média de todos pares formados entre os *spels* em V_m e os seus vizinhos de borda, e a_m é a média e b_m é o desvio padrão para

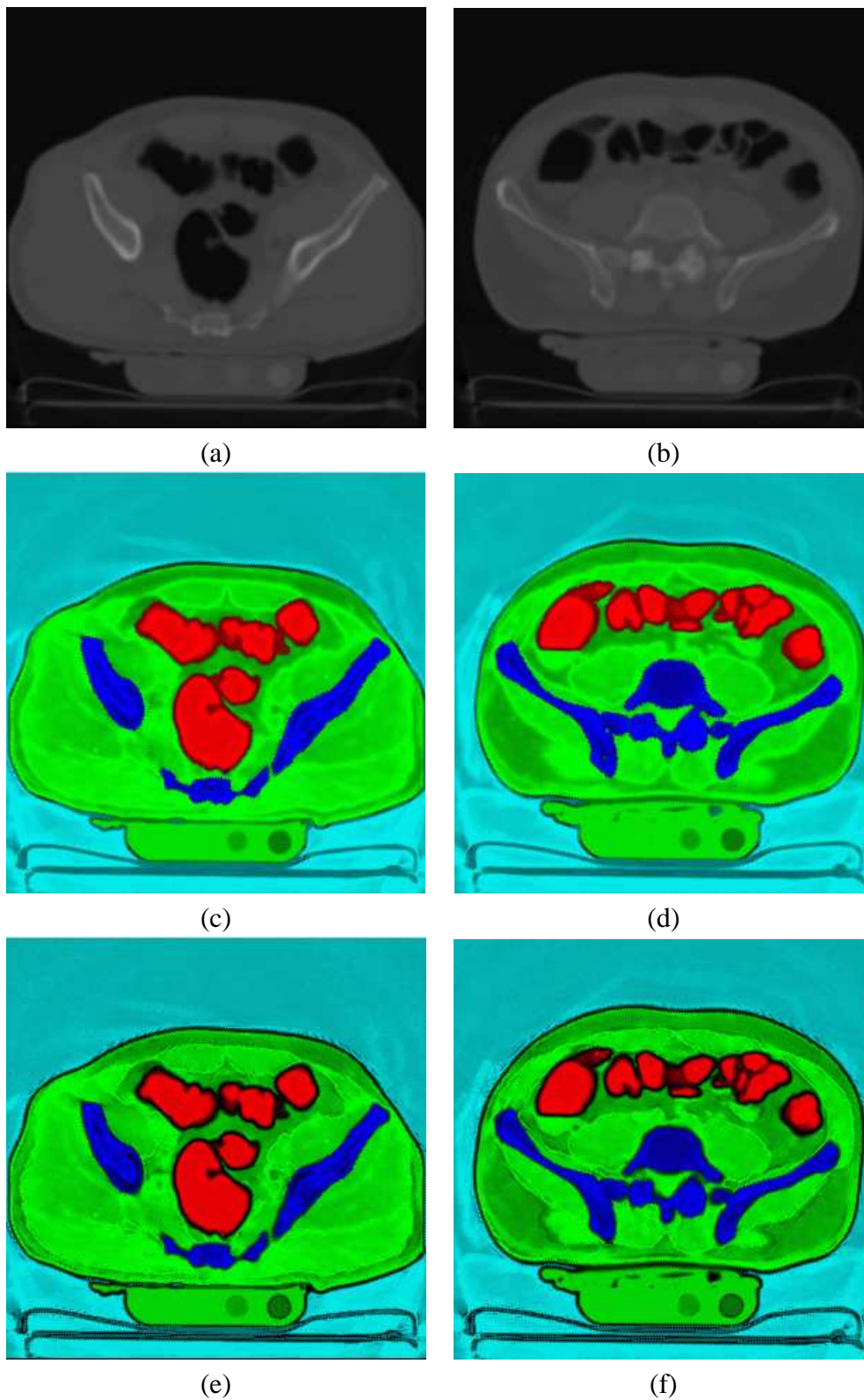


Figura 4.4: Comparação do algoritmo MOSFS original e a versão rápida. Duas fatias de um volume de tomografia computadorizada (a-b) e as suas correspondentes 4-segmentações obtidas utilizando o algoritmo original MOFS (c-d) e versão rápida do MOFS (e-f) (Imagens obtidas de Carvalho et al. [CHK05]).

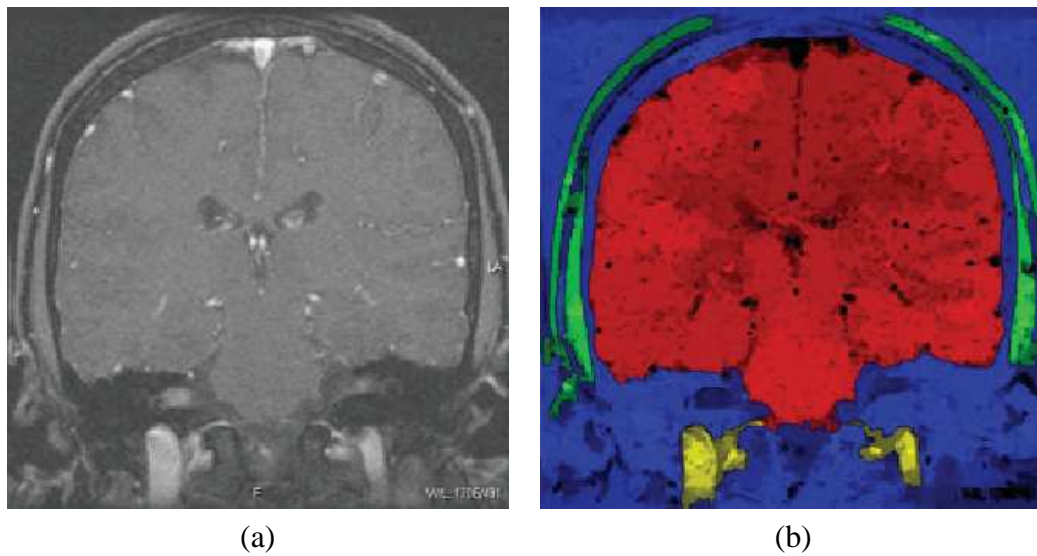


Figura 4.5: Exemplo de segmentação fuzzy. Uma imagem de ressonância magnética de um crânio (a) e a 4-segmentação deste crânio (b) (Imagens obtidas de Carvalho et al. [CHK05]).

todas as diferenças absolutas de intensidade (valores dos *spels*) entre os *spels* em V_m e seus vizinhos de borda. O Valor de g e a se referem a média e a diferença absoluta entre as intensidades de c e d respectivamente. A função $\rho_{r,s}(x)$ é uma função de densidade de probabilidade gaussiana com média r e desvio padrão s multiplicada por uma constante para que o seu valor máximo seja 1.

O primeiro exemplo é a segmentação de uma imagem de ressonância magnética de uma cabeça. Nesse exemplo é feita uma 4-segmentação de uma cabeça, onde para cada parte que se deseja identificar são escolhidas sementes para representar as características a serem mapeadas. O resultado da segmentação é visto na Figura 4.5.

4.5.1 Seleção de Partículas em Micrografos

A reconstrução 3D de macromoléculas é uma importante ferramenta no estudo do comportamento das mesmas, pois a sua estrutura 3D (forma) da macromolécula pode conter informações essenciais sobre o seu comportamento [ZMSH03]. Essas estruturas tridimensionais podem ser reconstruídas a partir de projeções 2D das macromoléculas que podem ser obtidas através da microscopia óptica ou eletrônica. Para a reconstrução de um modelo 3D de macromoléculas com a dimensão de 10 a 15 Å (10^{-10} m), podem ser necessárias entre 10.000 e 100.000 projeções deste espécime [FWM⁺02]. A Figura 4.6 contém uma sub-imagem (200×200 pixels) de um micrografo gerado a partir de um microscópio eletrônico.

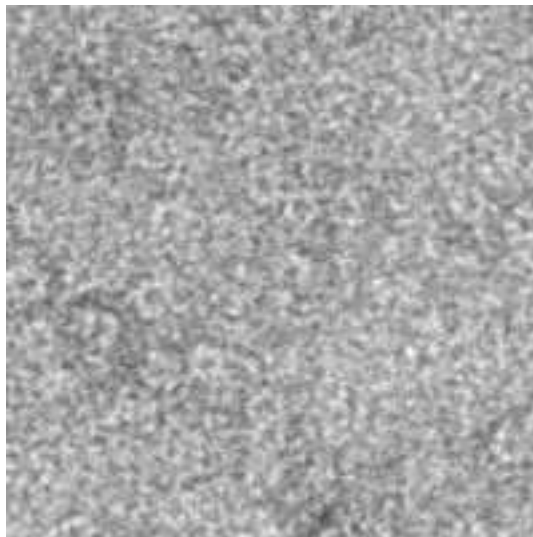


Figura 4.6: Sub-imagem de um micrografo. Uma sub-imagem (200×200 pixels) retirada de um micrografo obtido através de um microscópio eletrônico (imagem retirada de Carvalho et al. [COG06])

A identificação e a seleção de partículas nos micrografos gerados pela Microscopia Eletrônica (EM) tem se apresentado como um sério desafio para o processo de reconstrução de macromoléculas [Ros03]. As técnicas utilizadas para obtenção desses micrografos, descritas em Heel [vHGM⁺00], produzem imagens com baixos valores de SNR (*Signal to Noise Ratio*), dificultando assim a identificação das partículas nos micrografos (Figura 4.6). Essa figura apresenta uma janela de 200×200 pixels retirada de um micrografo obtido através de um microscópio eletrônico. Pode-se observar a dificuldade de se distinguir as partículas do background.

Em Carvalho et al. [COG06] foi apresentado um procedimento semi-automático para seleção de partículas em micrografos que reduz a taxa de falsos positivos (número de não partículas que são identificadas erroneamente como partícula no procedimento de identificação quando comparado com uma seleção feita manualmente). Quanto menor for a taxa de falsos positivos maior será a confiabilidade da reconstrução da estrutura 3D das macromoléculas.

O procedimento que foi desenvolvido para seleção das partículas nos micrografos pode ser dividido em três etapas: pré-processamento, correlação e refinamento. O fluxograma detalhado do funcionamento do procedimento pode ser visto na Figura 4.7. O primeiro passo da etapa de pré-processamento é a redução do tamanho original do micrografo (2048×2048 pixels) em uma escala de 1:8. Isso é feito para diminuir a área a ser pesquisada e melhorar a visualização das partículas no micrografo, eliminando assim parte do ruído. Depois são retirados 10 exemplos de partículas que serão utilizadas

para construção de um template com a média das intensidades em cada um desses exemplos. Em seguida, foi aplicado no micrografo reduzido o operador Sobel [GW02] para realçar as bordas das partículas existentes no micrografos. O último passo na etapa de pré-processamento é a aplicação de uma 1-segmentação *fuzzy* para identificar a região do background da imagem, ou seja, identificar quais regiões da imagem não contém partículas.

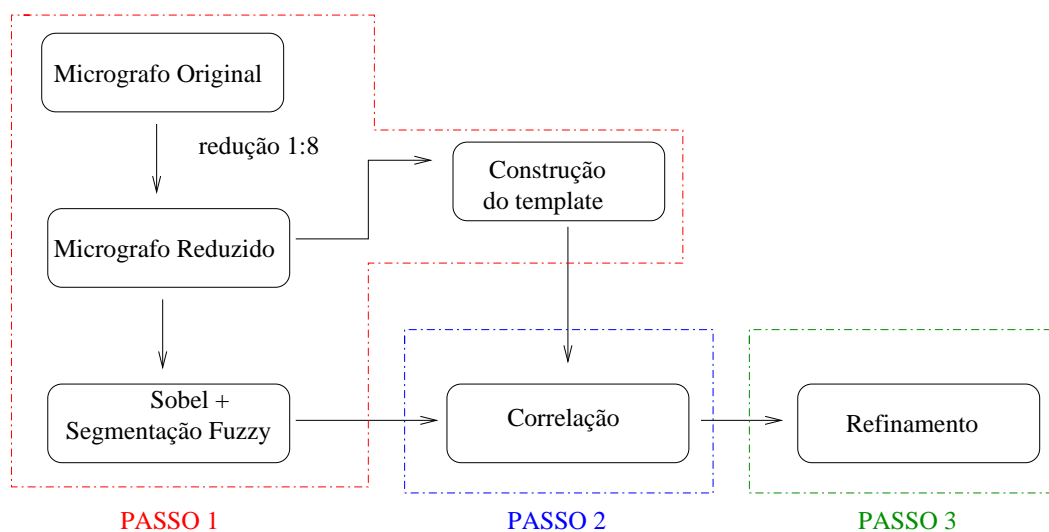


Figura 4.7: Fluxograma detalhado do funcionamento do procedimento para seleção de partículas em micrografos.

Para aplicar a segmentação fuzzy no micrografo, foram selecionados manualmente alguns pixels sementes para a região background do micrografo. A função de afinidade utilizada para calcular a pertinência de um pixel ao background é definida de acordo com a Equação 4.3. O resultado desse pré-processamento pode ser visto na Figura 4.8 (b). Nessa imagem, quanto mais perto de zero for a intensidade de um pixel, ou seja, mais perto do branco, maior será a probabilidade do pixel pertencer ao background do micrografo.

Na segunda etapa do processamento - a correlação - o resultado da segmentação *fuzzy* foi utilizado para construir um mapa que servirá para indicar em quais pontos deverá ser aplicada a correlação na imagem original do micrografo. A construção desse mapa foi baseada no grau de pertinência que cada pixels têm com o background. Somente nos pixels que tiverem um grau de pertinência menor que 0.4, ou seja, pouca probabilidade de pertencer ao background, a correlação vai ser aplicada. A Figura 4.8 (c) mostra a imagem do micrografo segmentado após a aplicação de um limiar (0.4) para isolar os pixels que vão sofrer a correlação (a região branca da imagem é onde deverá ser aplicada a correlação).

Com a utilização desse mapa é evitado um esforço computacional para calcular a

correlação em pixels que provavelmente não são pixels pertencentes a partículas. Uma outra vantagem constatada com a utilização do mapa de afinidade foi a redução da taxa de falsos positivos. Isso acontece devido ao fato que partes do micrografo que possuem uma baixa probabilidade não são analisados.

A correlação é uma técnica geralmente utilizada para encontrar ocorrência de uma sub-imagem $w(x, y)$ de dimensões $K \times L$ em uma imagem $f(x, y)$ de dimensões $M \times N$ onde $K < M$ e $L < N$. A correlação $c(x, y)$ entre as imagens $f(x, y)$ e $w(x, y)$ pode ser dada por

$$c(x, y) = \frac{1}{KL} \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} f_{i,j} w_{i,j}, \quad (4.4)$$

para $x = 0, 1, \dots, M-1$ e $y = 0, 1, \dots, N-1$, onde $f_{i,j} = f(i, j)$ e $w_{i,j} = w(x+i, y+j)$. Para cada par (x, y) , um valor de $c(x, y)$ é computado e o máximo valor de c indica a posição (x, y) onde w é mais semelhante a f . Uma deficiência nessa medida de correlação, a Equação 4.4, é a sensibilidade a troca de amplitude em f e w , e a dependência do valor de $c(x, y)$ do tamanho de w . Para resolver esse problema, uma função de correlação cruzada normalizada (*normalized cross-correlation*) [GW02] tem sido utilizada

$$c_N(x, y) = \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} \frac{[f_{i,j} - \bar{f}][w_{i,j} - \bar{w}]}{\sigma_f \sigma_w}, \quad (4.5)$$

onde \bar{f} e σ_f , e \bar{w} e σ_w são a média e o desvio padrão de $f_{i,j}$ e $w_{i,j}$ respectivamente. Uma vez que as imagens têm diferentes tamanhos, uma mascara binária $M_{i,j}$ pode ser introduzida para produzir uma função de correlação local [Ros03]

$$c_L(x, y) = \frac{1}{P} \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} \frac{[f_{i,j} - \bar{f}] M_{i,j} [w_{i,j} - \bar{w}]}{\sigma_{Mf} \sigma_w}, \quad (4.6)$$

onde P é o número de pontos não zero em $M_{i,j}$. Essa função de correlação cruzada local, que é utilizada nesse trabalho, tem sido utilizada em vários algoritmos para a seleção de partículas em micrografo utilizando *template matching*, como em Roseman [Ros03] e Rath [RF04].

Para aplicação da correlação cruzada normalizada foi utilizado um *template*, que foi construído no primeiro passo do pré-processamento, a partir da média de 10 exemplos de partículas que foram selecionadas manualmente no micrografo a ser analisado. O resultado da aplicação desse template no micrografo, que pode ser visto na Figura 4.8 (d), indica os possíveis centros das partículas nos micrografos. Os pixels com valores de correlação maiores que 0.4 (numa faixa entre $[-1,1]$) são selecionados como possíveis

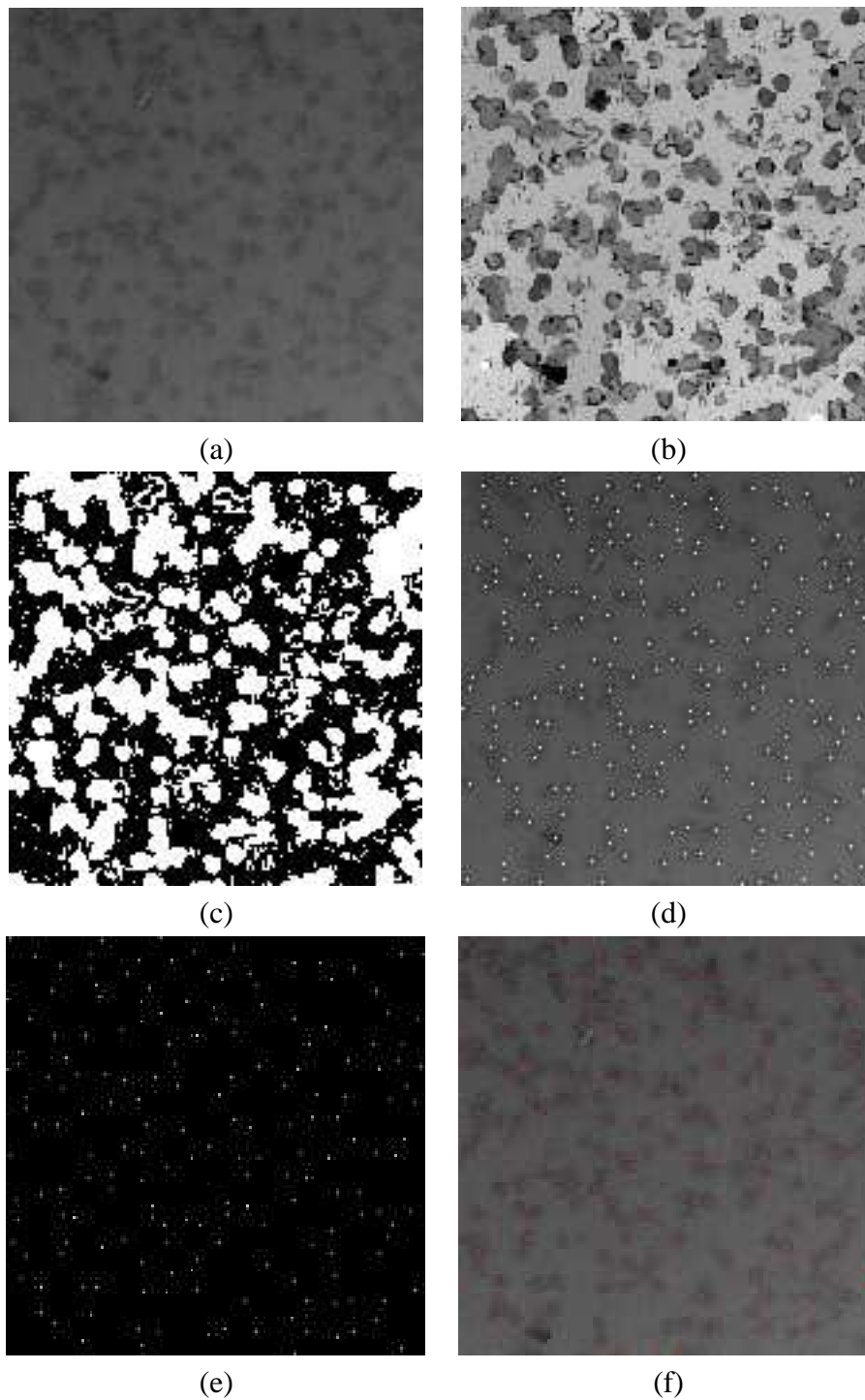


Figura 4.8: Imagens do processo de identificação e seleção de partículas no micrografo. Sub-imagem de um micrografo original (a), imagem pré-segmentada ou mapa de afinidade (b), threshold do mapa de afinidade (c), centro das partículas selecionadas por correlação (d), o micrografo depois da etapa do refinamento (e) e o micrografo com a seleção manual (f). (Imagens extraídas de [COG06]).

centros das partículas.

Como última etapa do procedimento, os pontos vizinhos que foram detectados como centros de partículas são unidos para gerar um único centro de partículas. Os pixels isolados que estão localizados perto de um possível centro de partículas a uma distância menor que o diâmetro médio das partículas, são eliminados. O resultado da etapa do refinamento pode ser vistos na Figura 4.8 (e).

Para analisar o desempenho do método proposto, foi realizada uma comparação deste com uma seleção de partículas feita manualmente, onde um falso negativo (FN) é encontrado se não existir um centro de partícula selecionado pelo método proposto cuja menor distância para um centro de partícula selecionado manualmente seja menor que a metade da largura da partícula. Um FP (falso positivo) é encontrado se não existir um centro de partícula selecionado manualmente cuja menor distância para um centro de partícula selecionado pelo procedimento proposto seja menor que a metade da largura da partícula.

	FP	FN
Faixa aceitável para a seleção de partículas	10%	25%
Método proposto sem o mapa de afinidades	10,50%	19,82%
Método Proposto com o mapa de afinidades	5,55%	23,23%

Tabela 4.1: Quantidade de falsos positivos e falsos negativos para seleção de partículas em micrografos.

De acordo com Zhu [ZCG⁺04], os métodos de seleção de partícula automático ou semi-automático mais utilizados possuem resultados na faixa de 10% para FP e 25% para FN. Os resultados obtidos utilizando a correlação em todo o micrografo analisado, ou seja, sem utilizar o mapa de afinidade construído a partir da 1-segmentação *fuzzy*, foram de 10,50 % de FP e 19,82 % de FN (ver Tabela 4.5.1). Apesar desses valores estarem próximos dos resultados dos métodos mais utilizados, o resultado obtido com a utilização do mapa de afinidade para guiar a aplicação da correlação, reduziu quase pela metade a taxa de FP. Com a utilização do mapa produzido pela segmentação *fuzzy* foram obtidas as taxas de 5,55 % para FP e 23,23 % para FN. Isso pode ser explicado pelo fato de que o mapa de afinidade evita o erro de análise em partes ruidosas do micrografo. Mesmo com o aumento da taxa de FN de 19,82 % para 23,23 % com a utilização do mapa (que ainda é menor que os 25% proposto por Zhu [ZCG⁺04]) o método de seleção apresentado possui resultados bastante satisfatórios.

Capítulo 5

Segmentação Fuzzy de Vídeos

A segmentação fuzzy de vídeos utilizará uma função de afinidade para possibilitar a expansão, a partir dos pixels sementes, das regiões que se deseja segmentar. Um dos objetivos da utilização da segmentação fuzzy para segmentar vídeos, é a análise da robustez deste algoritmo em vídeos que estejam sobre a influência de ruídos e/ou de uma iluminação não homogênea, já que este algoritmo de segmentação fuzzy apresenta bons resultados na segmentação de imagens com essas particularidades.

Para realizar a segmentação dos vídeos foram propostas duas metodologias: por propagação de sementes e pela análise do vídeo como um volume 3D. A primeira metodologia é baseada na propagação de pixels sementes, adicionados no primeiro frame, para os demais frames da seqüência de imagens. Essa metodologia apresenta uma aplicação limitada e será apresentada com mais detalhes na Seção 5.1. A segunda metodologia, que será apresentada na Seção 5.2, trata os vídeos como um volume 3D, e os seus resultados se mostram mais robustos tanto em vídeos sintéticos quanto em vídeos reais.

5.1 Segmentação Fuzzy de Vídeo por Propagação de Sementes

Esta metodologia de segmentação de vídeo baseia-se no algoritmo de segmentação fuzzy apresentado originalmente por Herman e Carvalho [HC01], e que foi descrita detalhadamente na Seção 4.2. O método proposto consiste na segmentação individual dos frames de um vídeo através da propagação dos pixels sementes que foram selecionados pelo usuário no primeiro frame de um vídeo.

Através de uma interface gráfica, o usuário seleciona no primeiro frame da seqüência os pixels sementes para cada objeto a ser segmentado. Após essa seleção, o usuário po-

derá visualizar o resultado da segmentação fuzzy do primeiro frame antes da segmentação completa do vídeo. Caso o resultado da segmentação do primeiro frame não seja satisfatório, ele poderá adicionar e/ou remover pixels sementes dos objetos e obter uma nova segmentação. Depois do primeiro frame ser segmentado satisfatoriamente, a segmentação dos próximos frames poderá ser realizadas através da propagação das sementes. Esse artifício de visualizar a segmentação antes de segmentar todos os frames pode evitar que a segmentação produza resultados insatisfatórios.

A presente metodologia utiliza como função de afinidade a Equação 4.3, e foi inicialmente modelada para realizar a segmentação somente em vídeo em tons de cinza. A adaptação desta metodologia para a segmentação de vídeos coloridos pode ser realizada alterando-se o cálculo das funções de afinidade. Após a segmentação do primeiro frame, os pixels sementes de cada objeto são propagados explorando a coerência temporal entre frames do vídeo. Na prática, isto pode ser feito utilizando a diferença entre os frames nos pixels sementes, que mede a similaridade dos pixels baseando-se nos valores de suas intensidades. A diferença de intensidade entre os pixels sementes localizados na posição (x, y) para os frames n e $n - 1$ ($FD_{n,n-1}(x, y)$) [Tek95] pode ser obtida por:

$$FD_{n,n-1}(x, y) = S_c(x, y, n) - S_c(x, y, n - 1), \quad (5.1)$$

onde $S_c(x, y, n)$ e $S_c(x, y, n - 1)$ são as intensidades dos pixels (x, y) nos frames n e $n - 1$ respectivamente.

No procedimento de propagação dos pixels sementes, a diferença entre os frames n e $n - 1$ é calculada para todos os pixels sementes. Então, essa diferença é comparada com um limiar para decidir se esta semente continuará no próximo frame ou se deverá ser substituída automaticamente por uma nova semente. Para cada objeto m a ser segmentado, é definido um limiar L_m utilizando o valor do desvio padrão e da média dos níveis de cinza dos pares formados entre os pixels sementes e os seus vizinhos de borda. O objetivo da utilização desse limiar é tentar garantir que a posição do pixel semente no frame n pertença ao mesmo objeto do frame $n - 1$, evitando assim que ele não invada a região de outro objeto ou caia em uma região de borda. Para cada pixel semente (x, y) , a Equação 5.2 retornará 1 se o pixel semente de um objeto necessita ser removido, e 0 se ele deve ser propagado para o próximo frame.

$$Z_{n,n-1}(x, y) = \begin{cases} 1, & \text{se } \|FD_{n,n-1}(x, y)\| > Lr_m, \\ 0, & \text{caso contrário,} \end{cases} \quad (5.2)$$

A Figura 5.1 ilustra o esquema da propagação de sementes adotado nessa metodologia. Uma vez determinado quais sementes do objeto m deverão ser substituídas, é

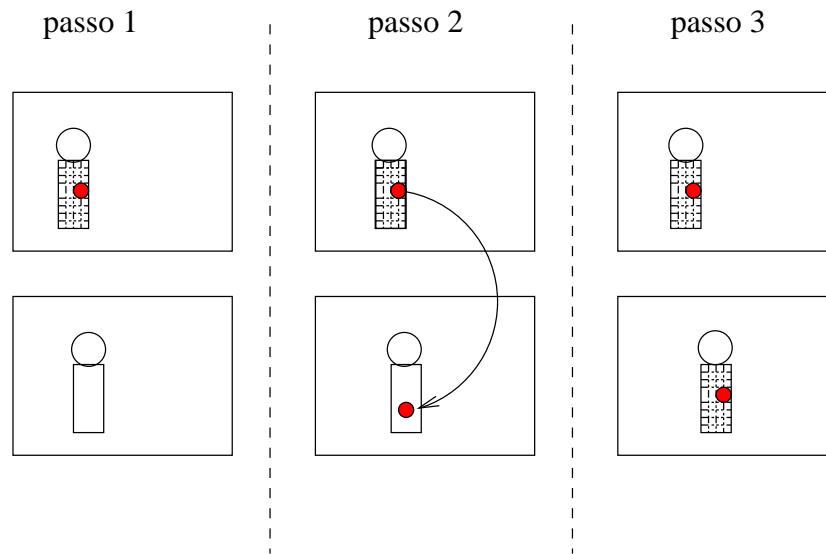


Figura 5.1: Funcionamento da segmentação fuzzy por propagação de sementes. Depois do primeiro frame ser segmentado, no Passo 1, a semente é propagada no Passo 2, e então o próximo frame é segmentado (Passo 3).

necessário selecionar novas sementes para serem adicionadas ao conjunto V_m . Para realização desse procedimento foi utilizada a seguinte heurística: após detectar que o pixel semente a do objeto m no frame k necessita ser substituído, todos os pixels rotulados por uma corrente originária de a que tiverem o valor da conectividade fuzzy maior que um limiar serão armazenados em um heap (os valores de conectividade serão utilizadas como chave do heap). Em seguida estes pixels são retirados um a um do heap para verificar a possibilidade da sua utilização como pixel semente. Para essa verificação foi utilizada a Equação 5.2. Caso nenhum pixel possa ser encontrado através desse procedimento, o pixel semente a será removido do conjunto V_m .

Com o decorrer dos experimentos foi verificado que apesar da segmentação fuzzy por propagação de sementes apresentar resultados satisfatórios para alguns vídeos, como no caso mostrado na Figura 5.2, este método não funcionou adequadamente resultados satisfatórios em algumas situações. Uma dessas situações ocorre quando as sementes propagadas para o próximo frame estão localizadas perto de regiões de fronteira entre dois objetos. Como essas regiões são de transição entre dois objetos que geralmente possuem uma alta variação nos valores dos pixels, as novas sementes podem alterar o cálculo da função de afinidade e assim haver perda de precisão na segmentação.

Uma outra situação ocorre quando o vídeo a ser segmentado é muito ruidoso. Neste caso, o problema também está na propagação da semente. Devido à presença do ruído, os pixels vizinhos da nova semente no frame n podem ter pouca ou nenhuma afinidade com esta semente, apesar desta nova semente ter uma alta conectividade fuzzy com a semente

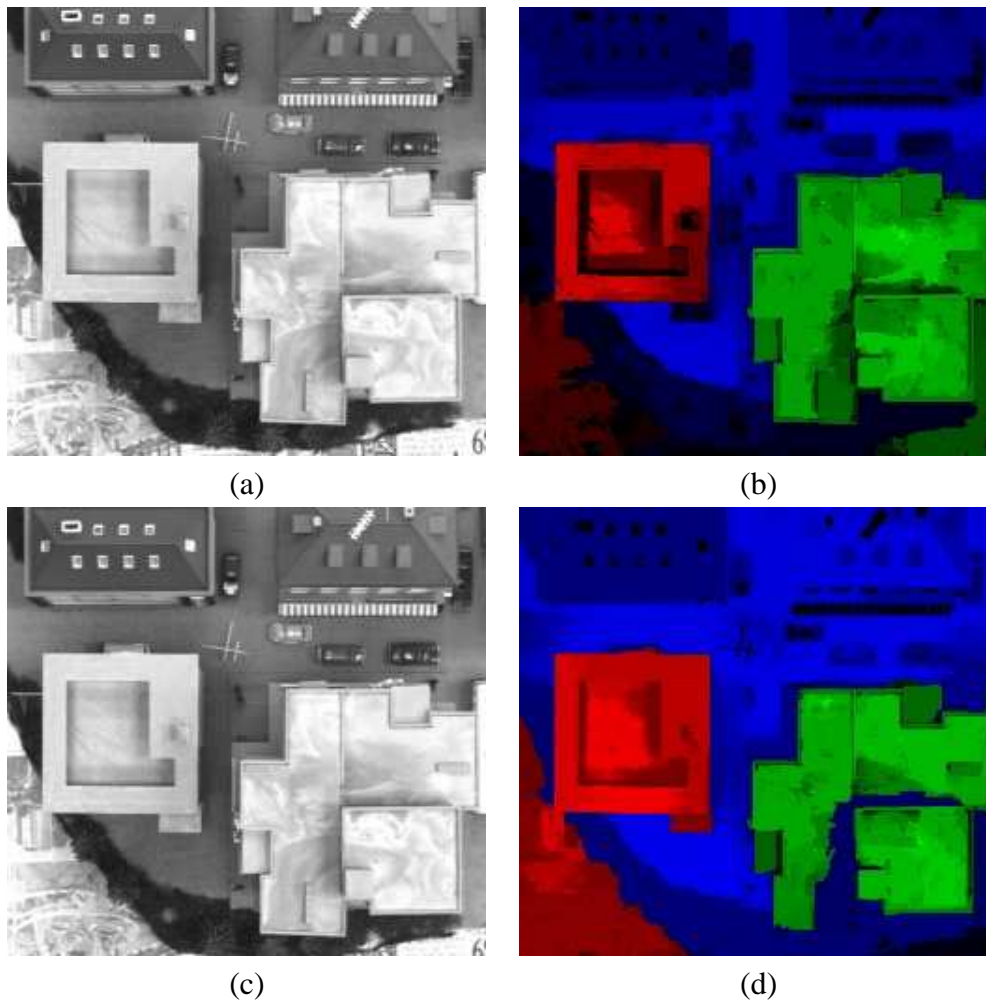


Figura 5.2: Segmentação fuzzy por propagação de semente. O primeiro e o sexto frame de um vídeo (original (a-c) e segmentado (b-d))

do frame $n - 1$ do qual ela foi propagada. A Figura 5.3 é um exemplo onde a proximidade dos pixels sementes nas regiões de fronteira afeta o cálculo da afinidade fazendo com que exista perda de precisão na segmentação.

A solução para esses problemas pode ser a utilização de outros mecanismos para a propagação das sementes de um frame para o outro. Uma alternativa para a solução desses problemas, é estimar o movimento dos pixels sementes através do fluxo óptico entre os frame $n - 1$ e n . Outra solução é proporcionar ao algoritmo de segmentação fuzzy a possibilidade de expandir as regiões não somente nas direções x e y , mas também na direção z . Na próxima subseção será apresentada uma metodologia de segmentação fuzzy de vídeo que utiliza informações de movimento e realiza expansões das regiões nos eixos x , y e z , que soluciona as deficiências vistas na segmentação fuzzy por propagação de sementes.

No resultado da segmentação fuzzy, representado por um mapa de conectividade, cada



Figura 5.3: Dificuldades na segmentação fuzzy por propagação de semente. Devido a alguns pixels se localizarem próximo às bordas, existe uma perda de precisão na segmentação (horizontalmente, verticalmente e em profundidade).

objeto é representado por um ou mais canais de cor do modelo *RGB*. Os valores atribuídos a esses canais correspondem ao grau de afinidade que o pixel tem com o objeto ao qual ele está rotulado. Por exemplo, se um objeto está representado pelo primeiro canal, o *R*, a cor $(255, 0, 0)$ será atribuída aos pixels que tiverem a máxima pertinência a esse objeto. Quanto menor for o valor atribuído a esse canal, menor será a afinidade que esse pixel terá com o objeto.

5.2 Segmentação Fuzzy de Volumes de Vídeo

Nesta subseção, será exposta uma metodologia para segmentação de vídeos baseada no método de segmentação fuzzy [HC01, CHK05] que calcula simultaneamente valores de pertinência para múltiplos objetos. A metodologia apresentada aqui considera um vídeo como um volume 3D, onde cada fatia z desse volume pode ser vista como os frames dessa seqüência (ver Figura 5.4 a). Como o objetivo desse trabalho não é segmentar vídeo em tempo real, e sim segmentar vídeos que foram previamente adquiridos, segmentar os vídeos como se fossem volumes 3D não representa uma restrição deste método.

Para realizar a segmentação em vídeos, que podem ser coloridas ou em tons de cinza, foi utilizado o algoritmo de segmentação fuzzy rápido descrito na Seção 4.4. Uma característica positiva advinda da utilização desse algoritmo de segmentação fuzzy rápido é que o usuário pode segmentar o volume (seqüência de imagens), analisar o resultado, adicionar e/ou remover pixels sementes, e então re-executar o algoritmo a fim de obter uma melhor segmentação em uma quantidade de tempo razoável.

A principal idéia da metodologia proposta é que o usuário possa interagir adicionando pixels sementes aos objetos em qualquer frame da seqüência, e através de uma função de afinidade realizar a expansão das regiões nas direções dos eixos x , y e z (Figura 5.4 b). A

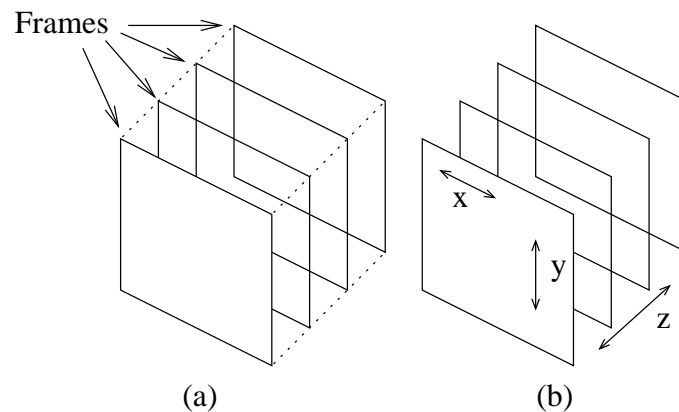


Figura 5.4: Um vídeo visto como um volume. O volume 3D (a) e as direções para onde as regiões podem ser expandidas (b).

vizinhança utilizada para crescimento das regiões foi adjacência de face (6-adjacência), que pode ser vista na Figura 5.5. A interação com o usuário acontecerá através do uso de uma interface gráfica, onde ele/ela poderá adicionar ou retirar os pixels sementes dos objetos em qualquer frame da seqüência.

Em algumas das segmentações realizadas nesse trabalho foram adicionada sementes somente no primeiro frame da seqüência. Mas a possibilidade da inserção de pixels sementes em qualquer frame além de aumentar o espaço amostral permitindo coletar informação dos objetos durante o decorrer da seqüência, fará com que objetos que não apareçam no início do vídeo possam ser segmentados.

Essa metodologia de segmentação fuzzy para segmentar vídeos como volume 3D foi apresentada em Carvalho et al. [COA06], onde foi utilizada somente informações de cor (dos canais Y, U e V do espaço de cor YUV) para a composição da função de afinidade. Apesar dos resultados obtidos serem satisfatórios na segmentação de alguns vídeos, existe

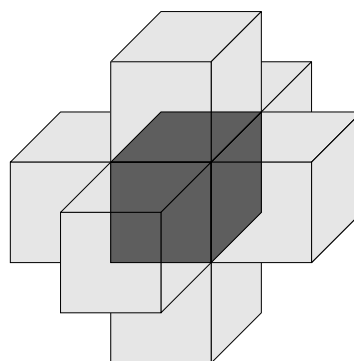


Figura 5.5: Vizinhança de face 3D. Voxel central (cinza escuro) e seus vizinhos adjacentes de face (cinza claro).

a possibilidade de aumentar a precisão da segmentação com a utilização de informações de movimento. A Figura 5.6 contém três frames de um vídeo segmentado utilizando a metodologia apresentada por Carvalho et al. [COA06].

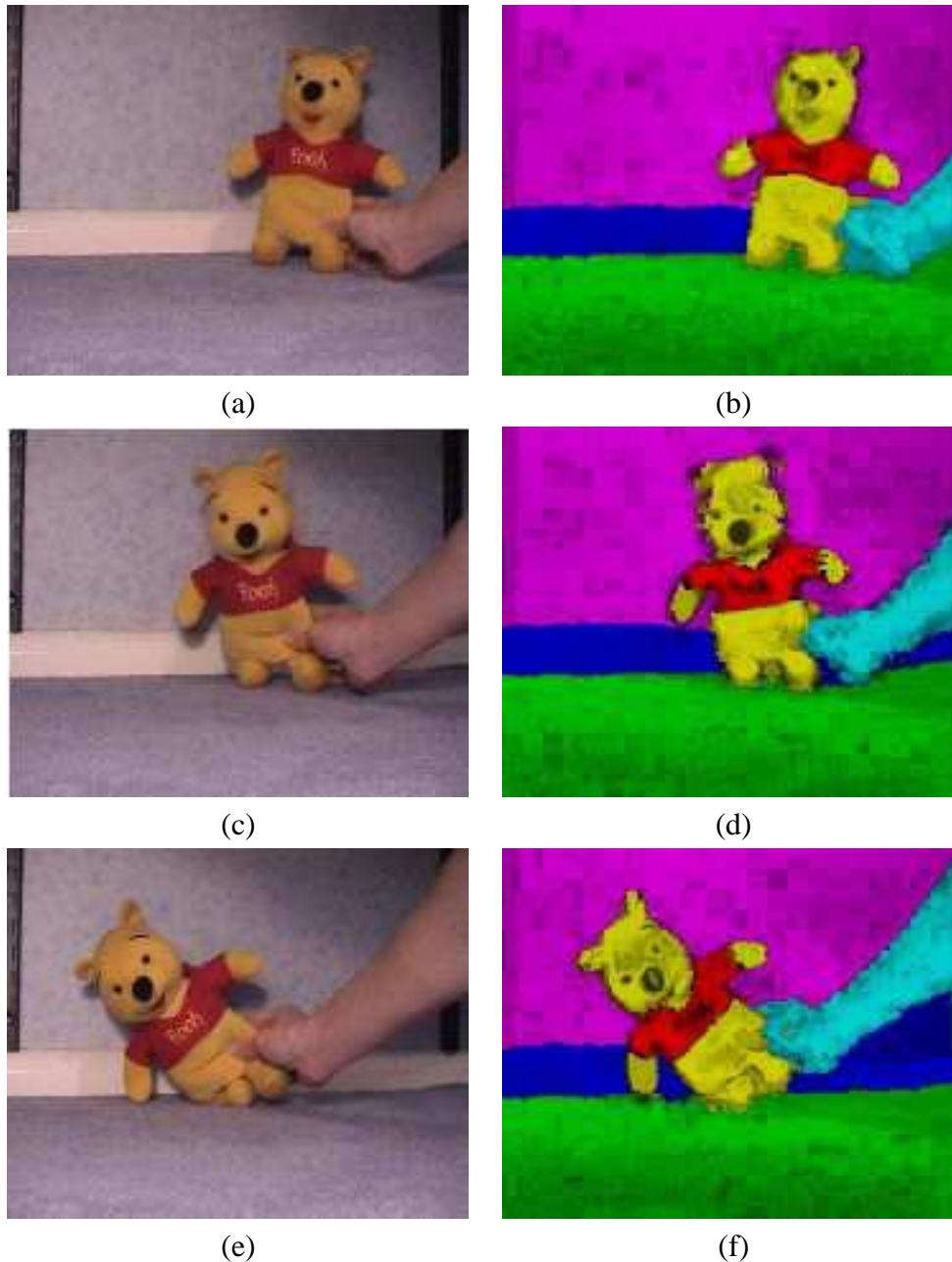


Figura 5.6: Exemplo de segmentação fuzzy de vídeo. Frames originais (a,c,e) e a segmentação fuzzy (b,d,f) de um vídeo. Cada linha corresponde a um intervalo de 15 frames.

No trabalho apresentado por Khan e Shah [KS01] é proposto um esquema que utiliza uma probabilidade a posteriori *maximum a posteriori probability (MAP)* framework com múltiplas características (como coordenadas espaciais, cor e movimento) para segmentar vídeos. Nessa abordagem, são atribuídos pesos para os atributos analisados buscando pri-

orizar ou penalizar as características que mais colaboraram para a segmentação do vídeo. A atribuição dos pesos é feita através da aplicação de um conjunto de heurísticas, que se propõe a estimar o erro relativo a estimação do movimento (ocasionado pela inconsistência do movimento e pela imprecisão da estimação nas bordas) e então atribuir mais ou menos importância a informação de cor ou de movimento. Uma das diferenças desta metodologia para a que está sendo apresentada nesta seção é que Khan e Shah utilizam um algoritmo de *clustering* para segmentar o primeiro frame da seqüência, e só então é realizada a propagação da segmentação para os demais frames.

O trabalho de Khan e Shah utiliza informações de cor e movimento para segmentar os vídeos. Isso é justificado, pois em alguns vídeos, os resultados da segmentação podem ser insatisfatórios quando a cor do objeto a ser segmentado é muito parecida com o *background* (ver Figura 5.7). Nessa situação, a informação de movimento pode auxiliar na distinção dos objetos a serem segmentados. Por outro lado, a segmentação utilizando somente informação de movimentos tem seus próprios problemas. Devido a problemas como oclusão de objetos, iluminação não uniforme e sombreamento, o fluxo óptico pode apresentar imprecisão nas bordas e em algumas partes dos objetos quando estes estão em movimento.

A metodologia proposta neste trabalho utiliza, além de informações de cor, informações de movimento para segmentar os vídeos. Para isto, é utilizada uma função de afinidade que é composta por três informações de cor e duas informações de movimento. As informações referentes a cor serão obtidas através da utilização de três canais (*ch1*, *ch2* e *ch3*) de um ou mais modelos de cores (cujos canais devem estar normalizados entre 0 e 1). Já as informações de movimento utilizadas na composição da função de afinidade, os vetores \bar{u} e \bar{v} , serão obtidas por meio do cálculo do fluxo óptico através da utilização do algoritmo de Proesmans [PGPO94] que foi discutido na Seção 3.4. Com o objetivo de diminuir o efeito da imprecisão na estimação do movimento foi utilizada somente a informação de magnitude do movimento na função de afinidade, descartando assim a informação de direção dos vetores do mapa de fluxo óptico.

A função de afinidade fuzzy é definida pela composição de 10 componentes, sendo dois componentes para cada informação analisada (*ch1*, *ch2*, *ch3*, \bar{u} e \bar{v}). Os dois componentes de cada uma dessas informações serão definidos usando a média e o desvio padrão da média de todos os pares formados entre os spels em V_m e os seus vizinhos adjacentes, e a média e o desvio padrão para a diferença absoluta entre todos os pares formados entre os spels em V_m e os seus vizinhos adjacentes. As funções parciais de afinidade de cor, Equação 5.3, e de movimento, Equação 5.4, serão definidas como 0 se os spels *c* e *d* não são adjacentes, e caso contrário, como

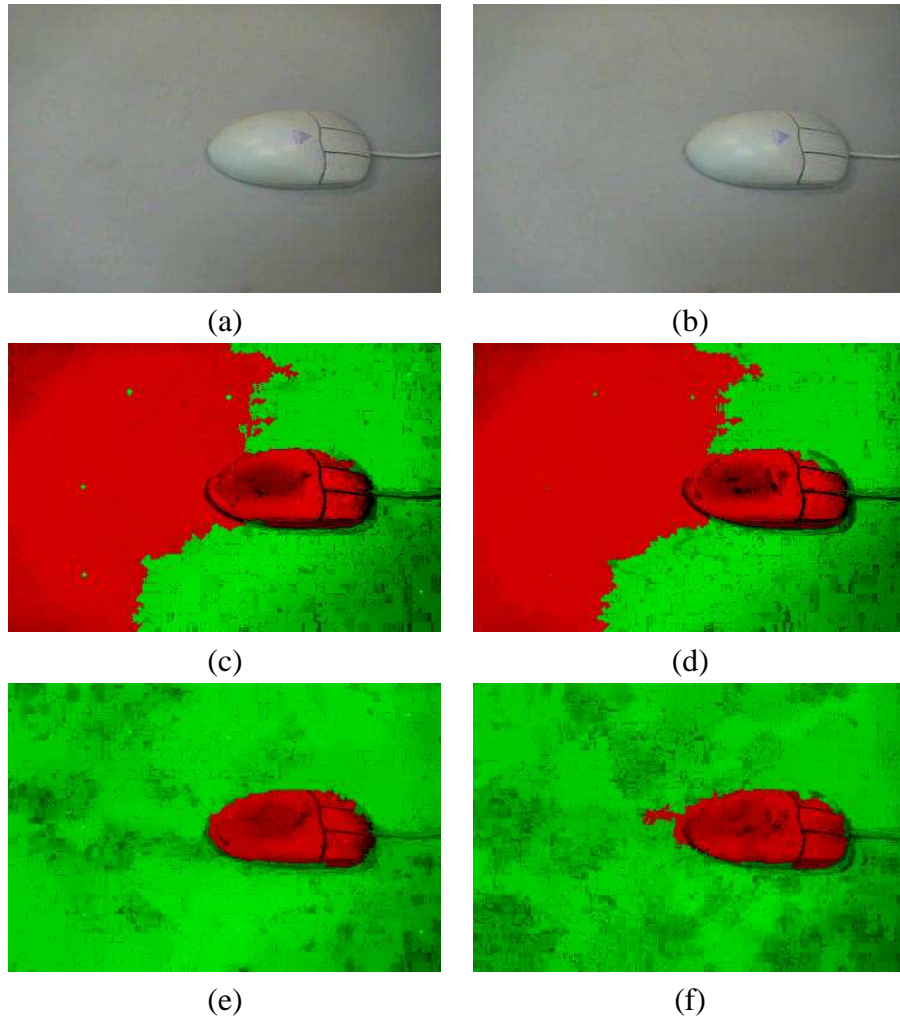


Figura 5.7: Utilização da informação de movimento para diferenciar o mouse do background. Frames originais (a-b), a sua segmentação fuzzy utilizando somente informações de cor (c-d) e com informações de cor e movimento (e-f).

$$\psi_m(c, d)_{cor} = \frac{\sum_{i=1}^3 (\rho_{g_{i,m}, h_{i,m}} + \rho_{a_{i,m}, b_{i,m}} g_i) / 2}{3} \quad (5.3)$$

$$\psi_m(c, d)_{vel} = \frac{(\rho_{g_{\bar{v},m}, h_{\bar{v},m}}(g_{\bar{v}}) + \rho_{a_{\bar{v},m}, b_{\bar{v},m}}(a_{\bar{v}}) + \rho_{g_{\bar{u},m}, h_{\bar{u},m}}(g_{\bar{v}}) + \rho_{a_{\bar{u},m}, b_{\bar{u},m}}(a_{\bar{v}}))}{4}, \quad (5.4)$$

onde $g_{i,m}$ é a média e $h_{i,m}$ é o desvio padrão para a média dos valores do canal i ($1 \leq i \leq 3$) de todos os pares formados pelos pixel sementes do objeto m e os seus vizinhos de borda, e $a_{i,m}$ é a média e $b_{i,m}$ é o desvio padrão para todas as diferenças absolutas dos valores do canal i para todos os pares formados entre os valores dos pixels sementes do objeto m e seus vizinhos de borda. Na Equação 5.4 as funções g , h , a e b tem a mesma

definição que tiveram para os canais de core, só que agora com os vetores do fluxo óptico \bar{u} e \bar{v} .

Para obter um melhor resultado na segmentação dos vídeos, são atribuídos pesos para cada uma das duas funções parciais de afinidade ($\psi_m(c, d)_{cor}$ e $\psi_m(c, d)_{vel}$). O objetivo dessa diferenciação nas informações que compõem a função de afinidade fuzzy é atribuir importância às informações que mais contribuem para a separação dos objetos na cena. A função de afinidade total é dada por

$$\psi_m(c, d) = w_1\psi_m(c, d)_{vel} + w_2\psi_m(c, d)_{cor} \quad (5.5)$$

onde w_1 e w_2 são pesos atribuídos às informações de cores e velocidade, respectivamente, e $w_1 + w_2 = 1.0$.

5.3 Seleção de Informações de Cores Não-Correlacionadas

Como já foi abordado nas Seções 2.2.1 e 2.2.2, dentre os diversos modelos utilizados para representar informações de cor em imagens, alguns proporcionam uma melhor separação entre os objetos existentes em uma cena do que outros modelos. A tarefa de testar cada modelo para encontrar os canais de cores que melhor segmentam um vídeo, pode ser custosa devido à existência de um grande número de espaços de cores a serem analisados.

Com o objetivo de aumentar a qualidade da segmentação fuzzy de vídeos, buscou-se encontrar uma metodologia que pudesse selecionar dentre um conjunto de modelos de cores, os canais que proporcionassem uma melhor segmentação dos objetos existentes na cena. Para isto, foi desenvolvido um pré-processamento que busca selecionar três canais de cores que possuam informações que sejam diferentes entre si, ou seja, deseja-se encontrar três canais de cores que possuam informações que não estejam correlacionadas. Com isto, pretende-se aumentar a quantidade e a variedade de informações que serão empregadas na função de afinidade utilizada no algoritmo de segmentação fuzzy.

Na Figura 5.8 (b,d,f), pode-se observar que nos canais R, G e B (do modelo *RGB*) contêm praticamente o mesmo tipo de informação. Já as informações dos canais H, S e I contêm uma maior variedade de informações, o que poderá contribuir para uma melhor segmentação. Provavelmente a utilização de um dos canais do modelo *HSI* (Figura 5.8 (c,e,g)) no lugar de um dos canais do modelo *RGB* proporcione uma melhor segmentação para esta seqüência.

De acordo com Hair [Hai05], o coeficiente de correlação de Pearson mede a intensidade ou o grau de associação entre duas variáveis, ou seja, mede a dependência linear

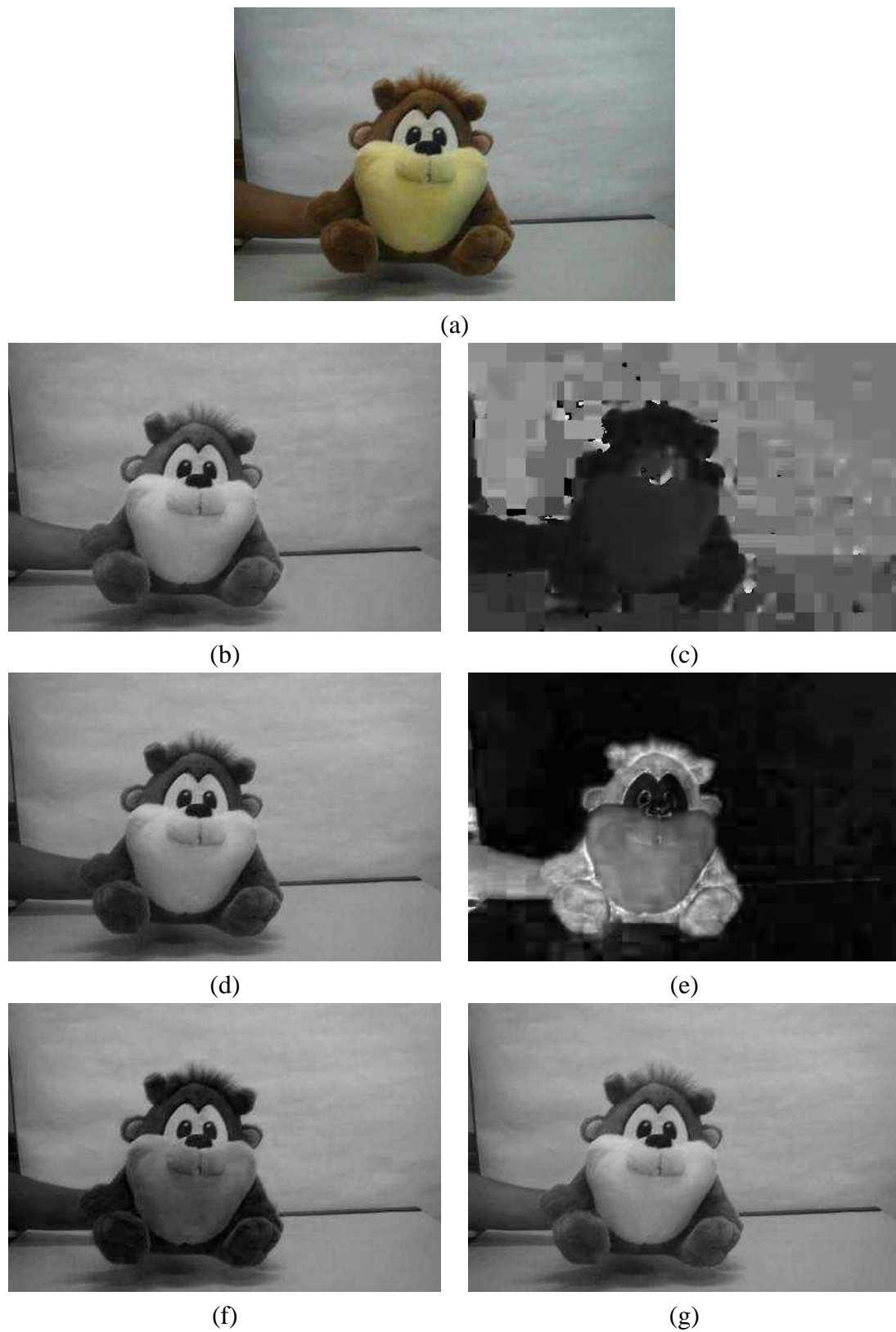


Figura 5.8: A imagem original do *Taz* (a) e a visualização isolada dos canais R (b), G(d), B(f), H (c), S (e) e I (g).

entre duas variáveis. Dadas duas variáveis, o coeficiente da correlação de Pearson informa um valor entre -1 e 1, sendo que 1 indica que as duas variáveis têm correlação perfeitamente positiva, ou seja, as variáveis apresentam a mesma distribuição linear. Se o valor obtido através da correlação for igual a -1 as variáveis apresentam correlação perfeitamente negativa, isto é, de acordo com a distribuição linear, se uma aumenta a outra diminui e se for igual a 0 as duas variáveis não apresentam correlação, ou seja, não existe dependência linear entre as duas variáveis. A correlação de Pearson pode ser definida como

$$\Omega_{i,j} = \frac{\sum_{t=1}^n (\pi_t^i - \bar{\pi}^i)(\pi_t^j - \bar{\pi}^j)}{\sqrt{\sum_{t=1}^n (\pi_t^i - \bar{\pi}^i)^2} \cdot \sqrt{\sum_{t=1}^n (\pi_t^j - \bar{\pi}^j)^2}}, \quad (5.6)$$

onde π_t^i e π_t^j são os valores que o pixel t possui para o canal i e j respectivamente, e o $\bar{\pi}^i$ e $\bar{\pi}^j$ são as médias dos valores de todos os pixels para os canais i e j . O valor de n é referente a quantidade de pixels sementes que foram selecionados pelo usuário. A matriz Ω conterá os valores das correlações entre todos os k canais analisados. Neste pré-processamento, pretende-se utilizar a correlação de Pearson para analisar as informações de cor dos pixels sementes em um conjunto de modelos de cores, e então escolher três canais que contenham o menor grau de associação entre eles.

Para encontrar esses três canais, foi utilizado o seguinte procedimento: o primeiro canal selecionado, aqui denominado de *ch1*, será o canal que tiver a menor correlação com todos os outros canais. O segundo canal a ser selecionado, o canal *ch2*, deverá ser o canal menos correlacionado com o primeiro canal selecionado. Por fim, será selecionado o terceiro canal, o *ch3* que é o canal menos correlacionado com os dois canais selecionados previamente [OPC⁺06]. A Figura 5.9 mostra os passos para a seleção dos canais utilizando a correlação de Pearson, onde a seta entre dois canais indica que estes canais possuem a menor correlação.

Para encontrar o valor do canal menos correlacionado com os outros canais, será calculado para cada canal o valor de Υ^i , que indicará quanto o canal i está correlacionado com os outros $(k - 1)$ canais analisados. O valor de Υ^i para o canal i pode ser encontrado por

$$\Upsilon^i = \sum_{j=1}^k |\Omega_{i,j}| - k, \quad (5.7)$$

onde o $\Omega_{i,j}$ é o valor da correlação entre os canais i e j , e k é à quantidades de canais analisados. Quanto maior for o valor absoluto de Υ^i , menos correlacionado com os outros $(k - 1)$ canais será o canal i [OPC⁺06]. O primeiro canal selecionado será o canal que tiver o maior valor de Υ^i , enquanto que o segundo canal selecionado será o canal i que

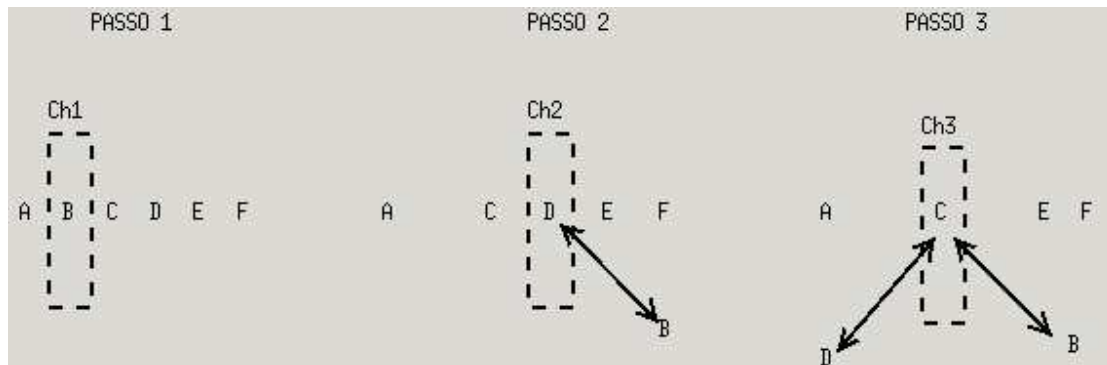


Figura 5.9: Seleção dos canais de cores utilizando a correlação de Pearson. As setas indicam as menores correlações achadas entre os canais.

tiver o menor valor absoluto na coluna $\Omega_{i,ch1}$ da matriz de correlação. Para selecionar o terceiro canal, será escolhido o canal i que minimize o valor de $\Omega_{i,ch1} + \Omega_{i,ch2}$.

5.4 Implementação

O estudo apresentado neste trabalho faz parte do projeto AnimVídeo, financiado pelo CNPq (PDPG-TI, processo 506555/06-6), que tem como principais objetivos a investigação de técnicas mais rápidas e robustas de segmentação e extração de movimentos em vídeo, além do desenvolvimento de técnicas de renderização não foto realista (*Non-Photorealistic Rendering* ou *NPR*) para produção de animações. Um dos objetivos da utilização da segmentação é possibilitar a renderização de objetos isolados em uma cena de vídeo.

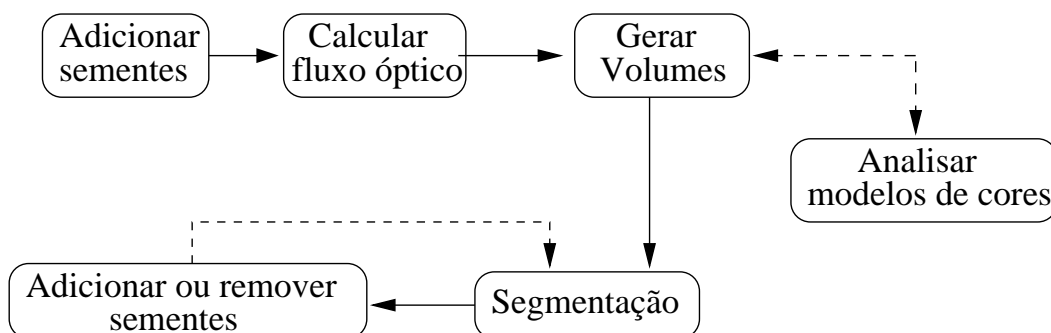


Figura 5.10: Fluxograma de funcionamento da segmentação fuzzy no *AVP Rendering*.

O software desenvolvido, denominado de *AVP Rendering*, consiste em uma ferramenta que permite ao usuário realizar segmentação em vídeo, e aplicar técnicas de ren-

derização em todo o vídeo ou em apenas alguns objetos presentes no vídeo. O programa, que ainda está em desenvolvimento, é composto por dois módulos básicos: módulo de segmentação e módulo de renderização. O usuário poderá interagir com os módulos do *AVP Rendering* tanto por uma interface gráfica quanto por linha de comando. O programa foi desenvolvido em C++ e QT 4.0 [Tro07, SJ06] utilizando uma plataforma Linux.

A segmentação de vídeos utilizando a ferramenta *AVP Rendering* consiste basicamente em quatro passos: escolher os pixels sementes para os objetos, calcular o fluxo óptico (opcional), escolher as informações de cores para gerar o volume a ser segmentado, e por fim executar a segmentação. A Figura 5.10 apresenta o fluxograma de funcionamento do módulo de segmentação de *AVP Rendering*. Depois de ser informado ao programa o frame inicial e o final de um vídeo, o usuário deverá selecionar as sementes para cada um dos objetos a serem segmentados. Para isto, o usuário deverá informar a qual objeto ele quer adicionar e/ou remover as sementes, e então utilizar o mouse para indicar a posição onde as sementes deverão ser adicionadas e/ou removidas.



Figura 5.11: Tela do *AVP Rendering*. Selecionando as sementes para segmentar um vídeo

A Figura 5.11 mostra um *screenshot* da tela inicial do *AVP Rendering*. Nesta figura, o número um indica a opção onde o usuário poderá salvar o conjunto de sementes selecionadas, e que futuramente poderão ser restauradas através da opção "load seed"(número 2 da Figura 5.11). Através da barra de rolagem, número 3 da Figura 5.11, o usuário poderá escolher qual o frame em que deseja adicionar ou remover as sementes, enquanto que através da opção "select object"(número 4), o usuário deverá indicar em qual objeto a semente deverá ser inserida. A ferramenta ainda possui um tocador de vídeo, número 5 da Figura 5.11, onde o usuário poderá ver a seqüência que será segmentada.

O segundo passo no processo de segmentação, que é o cálculo do fluxo óptico, pode ser feito utilizando a aba da segmentação fuzzy (ver Figura 5.12). Nesta opção, indicada na figura pelo número um, o usuário pode indicar os parâmetros necessários para a utilização do algoritmo de Proesmans desenvolvido por McCane [MNCG01], calcular o fluxo óptico, visualizar o mapa do fluxo óptico gerado e salvar/carregar um volume com os resultados da estimação do movimento para todos os frames do vídeo analisado. A opção de salvar o volume do fluxo óptico possibilitará que após o cálculo do mesmo para um vídeo, o usuário possa utilizar o fluxo óptico para segmentar novamente o vídeo quantas vezes ele desejar sem a necessidade de recalculá-lo.



Figura 5.12: Tela do *AVP Rendering*: Opção onde o usuário poderá segmentar o vídeo.

No terceiro passo, o usuário poderá gerar um volume do vídeo com os canais padrões utilizados pelo *AVP Rendering* (RGB , HSI e $I_1I_2I_3$), ou então gerar um volume utilizando três canais de qualquer modelo de cores utilizado pelo programa (número 3 da Figura 5.12). O modelo de cor padrão utilizado pela *AVP Rendering* é o RGB , mas o usuário poderá fazer uma análise das informações de cor utilizando a metodologia apresentada na Subseção 5.3 através da opção "*correlation*" (número 2 da Figura 5.12). O último passo é a aplicação da segmentação fuzzy no vídeo, que pode ser utilizada na opção indicada pelo número 4 da Figura 5.12.

5.5 Experimentos

Para analisar o desempenho do algoritmo de segmentação fuzzy em vídeos, foram utilizadas 6 vídeos. Os resultados das segmentações foram avaliados tanto qualitativamente (vídeos reais e sintéticos) como quantitativamente (vídeo sintético). A avaliação qualitativa é composta apenas por uma análise visual dos frames para uma determinada seqüência segmentada. Já na avaliação quantitativa, realizada apenas na seqüência sintética, foi utilizado um *ground truth*, e a precisão da segmentação será indicada pela quantidade de pixels rotulados corretamente. Também foi feita uma análise do desempenho do algoritmo sobre a influência de ruídos. Para isso foi adicionado um ruído gaussiano de média zero e desvio padrão 5 nos três canais de cores e em todos os frames do vídeo sintético.

Como o resultado das segmentações está diretamente associado a escolha das sementes para cada objeto, melhores resultados poderão ser encontrados para os mesmos vídeos analisados. Isso acontece porque o cálculo da afinidade entre os objetos existente nos vídeos é feito baseado no conjunto de pixels sementes selecionados pelo usuário. Todos os experimentos foram realizados em um computador Pentium IV 3.00 GHz com 2GB de memória RAM. Todos os vídeos que foram utilizados nesses experimentos, assim como os resultados das segmentações, estão disponíveis na URL <http://www.ppgsc.ufrn.br/~lmelo/>.

Atualmente o *AVP Rendering* está apto para executar a segmentação em vídeos analisando três modelos de cores: RGB , HSI e o $I_1I_2I_3$. Apesar dos experimentos terem sido feitos utilizando estes três modelos, essa metodologia pode ser ampliada para analisar outros modelos sem maiores restrições. O modelo RGB foi escolhido devido a sua larga utilização na segmentação de imagens e vídeos, e por ser um modelo utilizado por um grande número de dispositivos de captura de imagens e vídeos. A utilização do modelo HSI , que também é bastante utilizado na segmentação de vídeos, foi escolhido por ser um modelo que apresenta uma boa intuitividade com o sistema visual e porque é um modelo obtido por transformação linear do modelo RGB . Já o modelo $I_1I_2I_3$, que é

obtido por transformação não-linear do modelo *RGB*, foi escolhido por apresentar bons resultados na segmentação de imagens coloridas [CJSW01].

Vídeo	frames	Características
Taz	76	Movimento horizontal e vertical; ambiente interno; iluminação não uniforme; boa separação entre as cores do objeto; um único objeto se movendo 360 X 240 pixels
Demost	21	Movimento horizontal; ambiente interno; iluminação não uniforme; poucos detalhes; um único objeto se movendo 360 X 240 pixels
Caneca	70	Movimento horizontal e vertical; ambiente interno; pouco detalhe; 320 X 240 pixels
Sapo	85	Movimento horizontal, vertical e longitudinal; objetos com tons parecidas; iluminação não uniforme ambiente interno; dois objetos se movendo 360 X 240 pixels
Traffic Car	71	Movimento horizontal e vertical; vários objetos se movendo; ambiente externo; tons dos objetos parecidos 320 X 240 pixels
Office	20	Seqüência sintética; movimento de zoom 200 X 200 pixels

Tabela 5.1: Características dos vídeos utilizados nos experimentos.

Quatro dos seis vídeos que foram utilizados nesse trabalho foram obtidas pelos membros do projeto AnimVídeo utilizando uma câmera SONY Handicam DCR-HC90. O primeiro destes vídeos, denominado *Taz*, contém 76 frames e mostra um objeto se deslocando simultaneamente na horizontal e na vertical durante a cena. O segundo vídeo, chamado de *Demost*, mostra uma pessoa em um ambiente fechado (uma sala de aula) que passa em frente à câmera. Os últimos dois vídeos feitos pela equipe da AnimVídeo, *Sapo* e a *Caneca*, contêm respectivamente um sapo dançando e uma mão pegando uma caneca.

Os outros dois vídeos que foram utilizados em outros trabalhos sobre processamento e análise de vídeos. A primeira é uma seqüência de trânsito utilizada em controle de tráfego (denominada de *Traffic Car*) [KK03] e a segunda é um vídeo sintético que descreve um escritório [CVR07]. Os vídeos utilizados nos experimentos foram escolhidos com o objetivo de mostrar o efeito da segmentação em diferentes situações: ambiente externo (*Traffic Car*), ambiente interno (*Demost* e *Caneca*), vídeos sintéticos (*Office*) e vídeos para animação (*Taz* e *Sapo*). A Tabela 5.1 apresenta um resumo com as principais características dos vídeos utilizados nos experimentos.

A primeira análise que será feita é sobre a influência da utilização das características de cor e movimento um vídeo. Para isto foi utilizada a seqüência do *Traffic Car* (com

apenas três frames), onde serão rotulados quatro objetos: três automóveis e o background (asfalto e calçada). O mapa do fluxo óptico foi calculado utilizando como parâmetros lambda igual a 1000, número de iterações igual a 15 e 5 níveis. A Figura 5.13(a-c) mostra os três frames originais analisados, enquanto que a Figura 5.13(d-e), o mapa do fluxo óptico gerados entre esses frames.

Para mostrar a influência da informação de cor e movimento, foram feitas cinco segmentações com diferentes pesos para as informações de cor e movimento (ver Tabela 5.2). A Figura 5.14 mostra os resultados das segmentações para estas diferentes configurações. Na segmentação utilizando somente informações de cor (segmentação A da Figura 5.14), o objeto carro branco foi separado do asfalto devido à alta diferença entre as cores do carro e do asfalto. Já o objeto carro cinza, que absorve parte da calçada, e a caminhonete vermelha, que absorve boa parte da calçada e asfalto, têm alguma dificuldade em serem segmentados.

	Cor	Movimento
Segmentação A	100%	0%
Segmentação B	70%	30%
Segmentação C	50%	50%
Segmentação D	30%	70%
Segmentação E	0%	100%

Tabela 5.2: Configuração para segmentação do vídeo *Taz* para mostrar a influência da informação de cor.

Com a adição de informação de movimento na segmentação, a segmentação dos objetos carro cinza e caminhonete já é mais visível, enquanto que o carro branco continua com uma boa segmentação (ver segmentação B, C e D da Figura 5.14). Isso acontece devido a utilização da informação de movimento para separar o carro, que está em movimento, do objeto background que está parado.

Quando existem somente informações de movimento para realizar a segmentação (segmentação E da Figura 5.14), os objetos conseguem ser separados do background com uma certa redução na precisão da segmentação. Pode-se observar que apesar dos automóveis estarem separados do asfalto e da calçada, parte da forma dos objetos são perdidas nessa segmentação. Visualmente a melhor segmentação obtida foi a D, onde foram utilizadas 70% de informação de velocidade e 30% de informação de cor. Para esse experimento foram utilizados 37 pixels sementes (5 para o carro branco, 4 para o carro cinza, 5 para a caminhonete e 23 para o background).

A atribuição do peso para informação de cor e movimento depende da sequência a ser segmentada. É necessário analisar a estimativa do movimento para cada sequência,

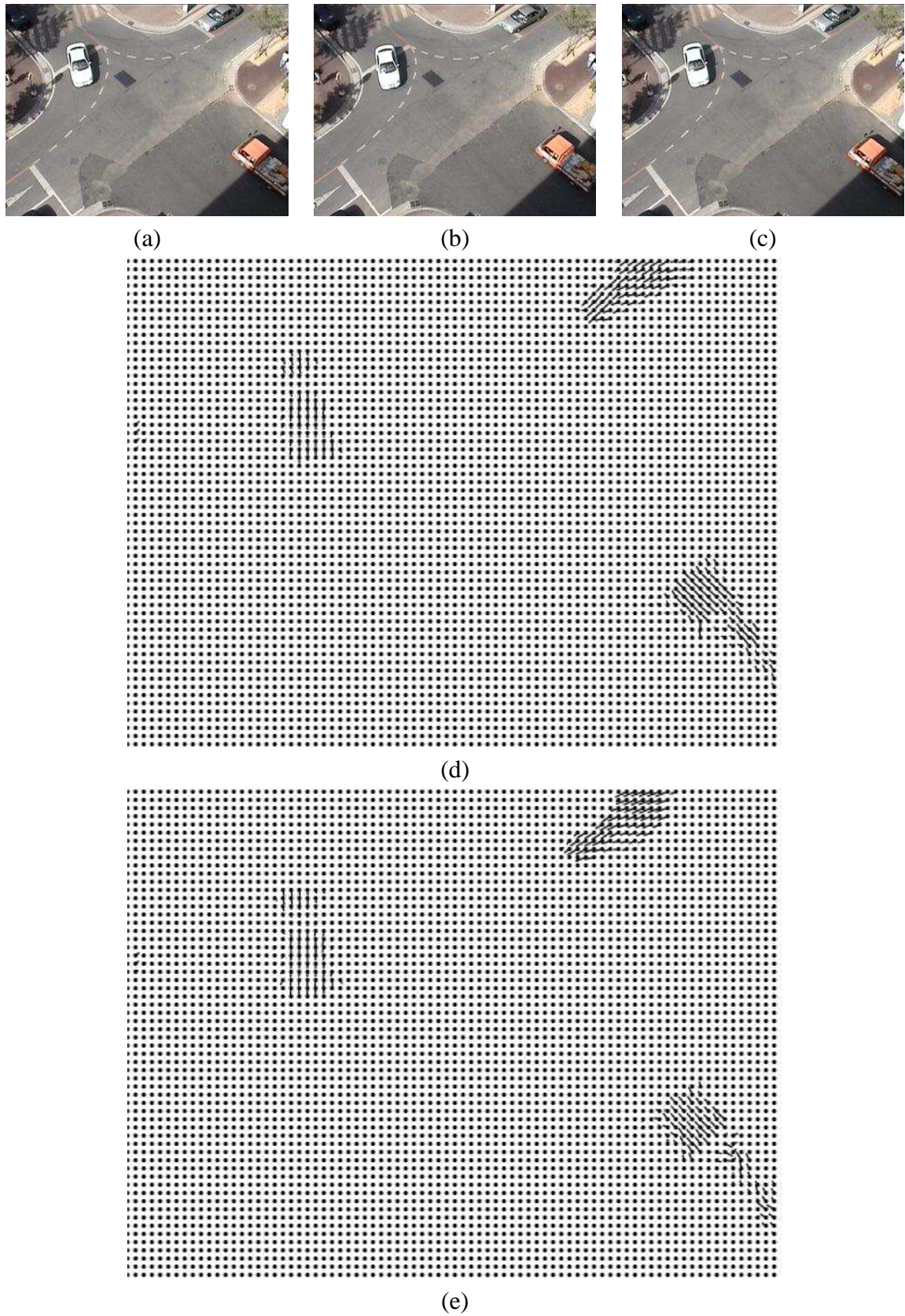


Figura 5.13: Fluxo Óptico gerado por três frames (a-c) do vídeo Traffic Car e os seus mapas do fluxo óptico.

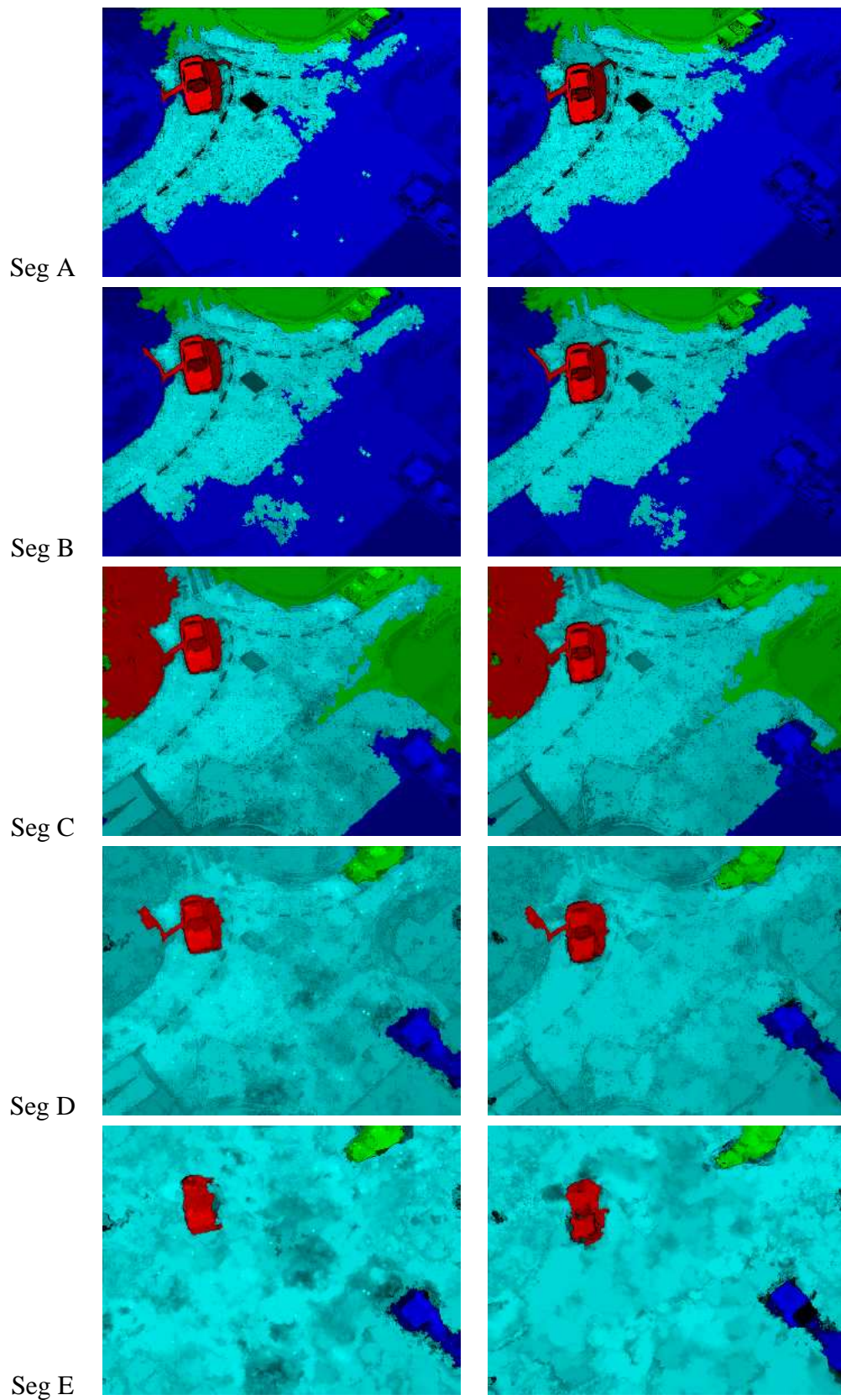


Figura 5.14: Resultado das segmentações do vídeo *Traffic Car* variando os pesos aplicados às informações e cor e movimento, usando os pesos descritos na Tabela 5.2.

e dependendo da precisão e da diferença dos vetores de fluxo óptico entre os objetos a serem segmentados, decidir qual peso atribuir a informação de cor e de movimento. A segmentação completa da seqüência do *Traffic Car* (Figura 5.15), que possui 75 frames, foi realizada utilizando 80% de informação de cor (modelo *RGB*) e 20% de informação de movimento. Nesta segmentação foram utilizados para calcular o fluxo óptico 5 níveis, 15 iterações e o lambda igual a 1000. A Tabela 5.3 mostra a lista dos objetos segmentados nessa seqüência.

Objeto	Cor	Descrição
1	vermelho	carro branco
2	verde	carro cinza
3	azul	caminhonete
4	ciano	asfalto
5	amarelo	calçada
6	magenta	sombra no passeio e no asfalto

Tabela 5.3: Configuração para segmentação do vídeo *Taz* para mostrar a influência da informação de cor.

Uma outra análise feita foi sobre o comportamento dos canais de cores selecionados pela metodologia proposta na Seção 5.3 para segmentação de vídeos. Em todos os seis vídeos analisados, o resultado da segmentação com os canais selecionados utilizando a correlação foi melhor que o pior resultado obtidos com os modelos de cores analisados. Em alguns vídeos, como por exemplo a do *Taz*, foi observado que a utilização dos canais selecionados na correlação apresentou melhores resultados que os obtidos com os modelos de cores *RGB*, *HSI* e $I_1I_2I_3$.

Na Figura 5.16 são apresentados os resultados das segmentações de três objetos na seqüência *Taz* utilizando os modelos de cores *RGB*, *HSI*, $I_1I_2I_3$ e os canais selecionados pela correlação I_3 , B e H. Para essa segmentação foram utilizadas 34 sementes selecionadas de uma só vez, ou seja, não foi selecionada mais nenhuma semente depois da primeira segmentação. Isso foi feito para que a inserção de novas sementes após a análise do resultado de uma das segmentações não influenciasse na qualidade dos resultados apresentados. Em uma análise visual em todos os 75 frames dessa seqüência, o resultado apresentado pela segmentação utilizando os canais selecionados pela correlação foi melhor que o obtido pelo modelo *HSI*, o modelo de cor que apresentou o melhor resultado. Nesta análise visual foi observada a precisão das segmentações nas bordas dos objetos e as partes dos objetos que foram classificada incorretamente.

Para fazer uma análise quantitativa dos resultados da segmentação foi utilizada uma seqüência sintética, e os seus resultados foram comparados com um groundtruth construído manualmente através da utilização do GIMP [GIM07]. A precisão da segmentação

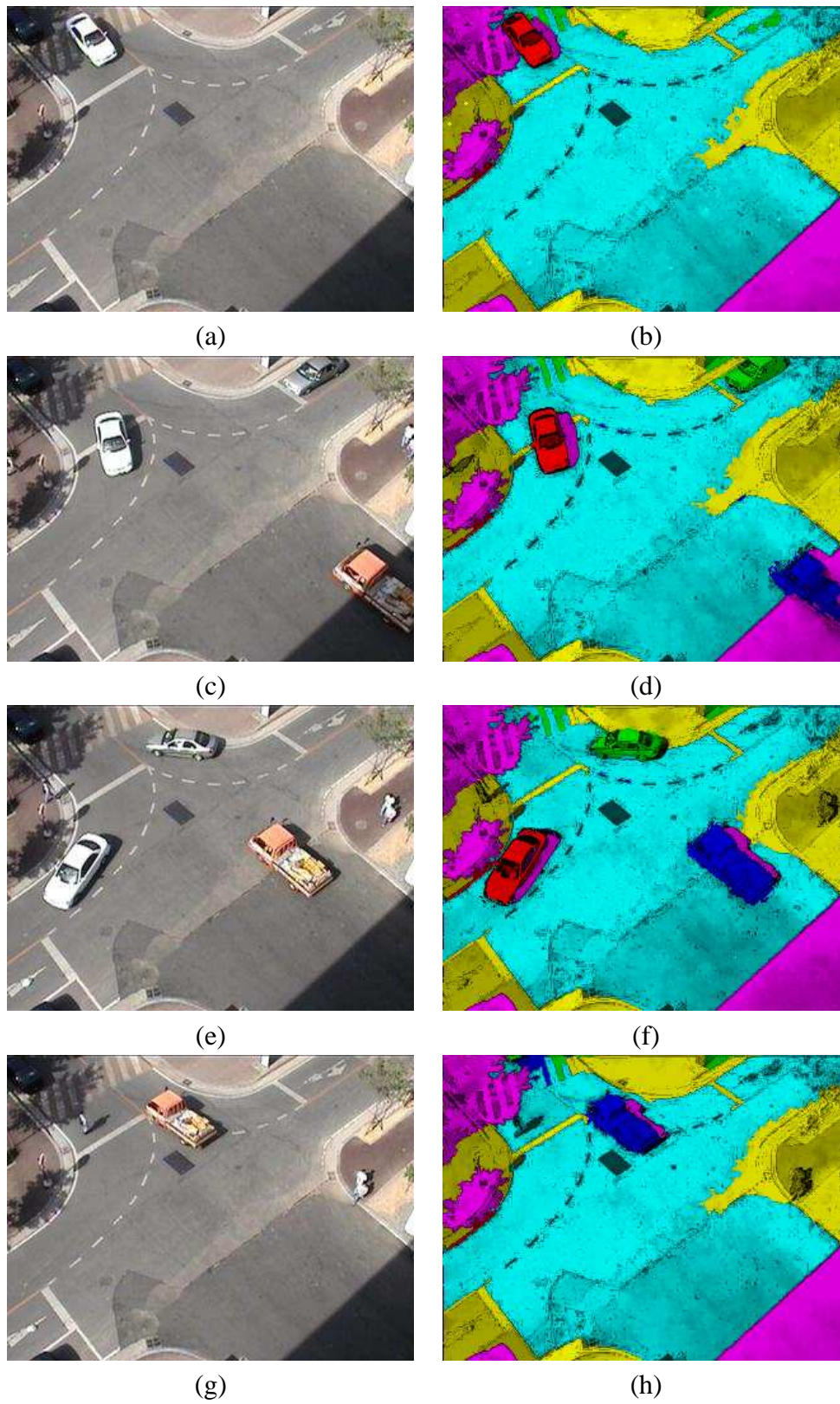


Figura 5.15: Resultado da segmentação do vídeo *Traffic Car* utilizando o modelo *RGB*. Cada linha corresponde a um intervalo de 20 frames.

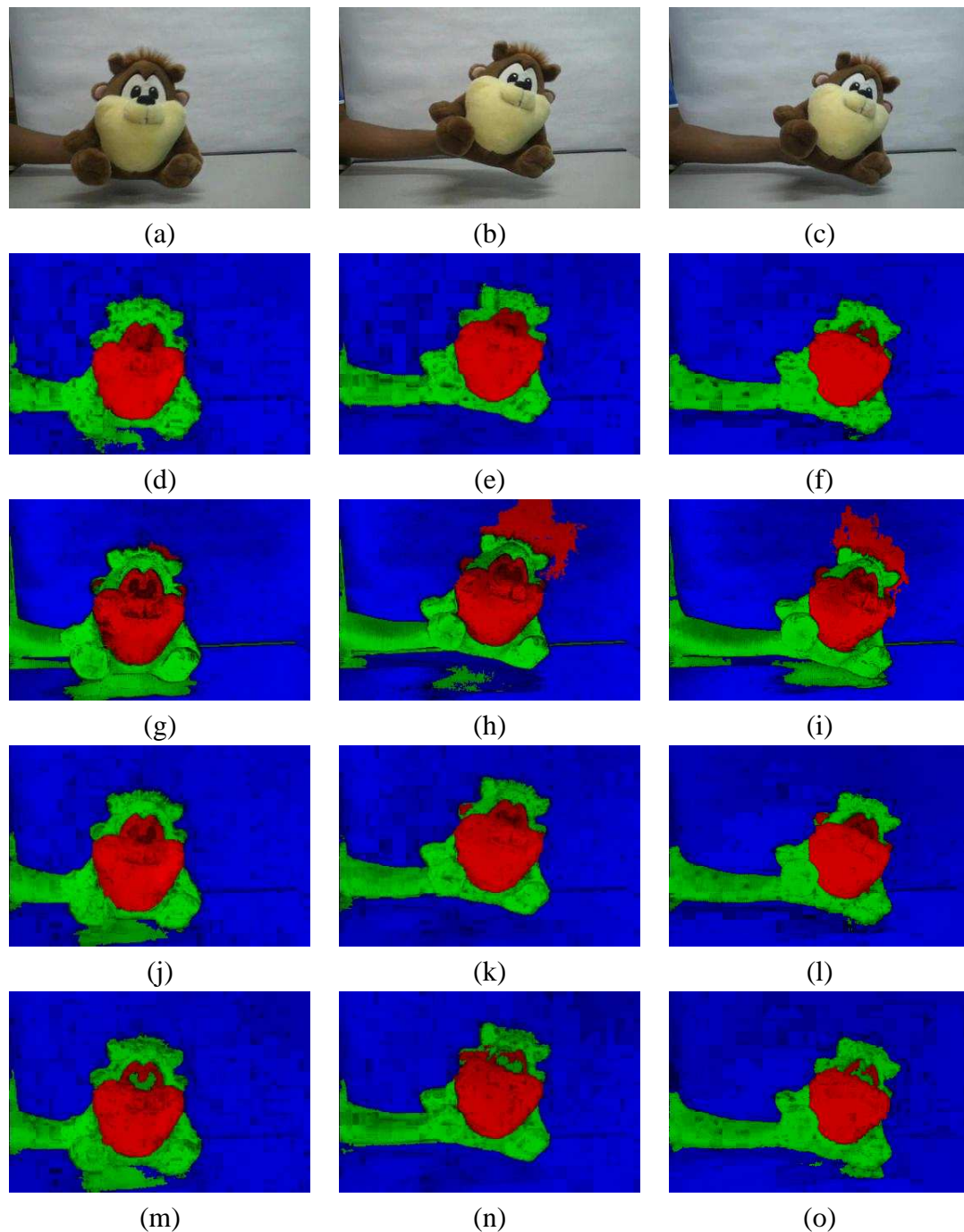


Figura 5.16: Resultados das segmentações do vídeo *Taz*, mostrando os frames 1, 30 e 70 segmentados com os canais I_3BH (d-f) selecionado pela metodologia da correlação de Pearson e pelo modelos de cores RGB (g-i), HSI (j-l) e os canais $I_1I_2I_3$ (m-o).

foi medida pela contagem do número de pixels classificados corretamente para todos os pixels da seqüência analisada. Para isto foram utilizados os 20 frames da seqüência *Office* e foram segmentados 7 objetos conforme a Tabela 5.4.

Objeto	Cor	Descrição
1	vermelho	parede
2	verde	piso
3	azul	cadeira
4	ciano	mesa, parte da armação do quadro e a caixa da janela
5	amarelo	computador
6	magenta	moldura do quadro e vidros da janela
7	preto	figura do quadro

Tabela 5.4: Objetos segmentados no vídeo do *office*.

Como os resultados das segmentações estão diretamente relacionados com a seleção das sementes, foram utilizados cinco conjuntos de sementes que foram selecionados por cinco voluntários (dois deles com relativa experiência na segmentação fuzzy de imagens e outros três com pouca experiência). Cada usuário teve uma única oportunidade para selecionar as sementes, ou seja, não foi permitido ao usuário excluir ou adicionar sementes após verificar o resultado das segmentações. A Figura 5.17 mostra o primeiro frame da seqüência sintética original (a) e com adição de ruídos (b).



(a)



(b)

Figura 5.17: Primeiro frame da seqüência sintética original (a) e com a adição de ruídos (b).

Os resultados das 40 segmentações realizadas na seqüência do *Office* são listados nas Tabelas 5.5 e 5.6. Para as segmentações sem adição de ruídos, os canais selecionados

	<i>RGB</i>	<i>HSI</i>	$I_1I_2I_3$	<i>Corr</i>
Usuário 1	86,15 ± 0,47	82,44 ± 0,54	94,14 ± 0,43	94,26 ± 0,36
Usuário 2	89,29 ± 0,57	93,66 ± 0,31	90,38 ± 0,50	93,54 ± 0,34
Usuário 3	87,02 ± 0,53	90,45 ± 0,37	88,80 ± 1,89	88,80 ± 1,89
Usuário 4	88,17 ± 0,45	92,94 ± 0,44	95,67 ± 0,43	95,18 ± 0,38
Usuário 5	88,19 ± 0,54	92,5 ± 0,35	90,08 ± 0,42	93,15 ± 0,67
Média	87,76 ± 0,51	90,40 ± 0,40	91,81 ± 0,72	93,15 ± 0,68

Tabela 5.5: Resultados das segmentações do vídeo sintético sem a adição de ruídos, onde os melhores resultados para cada seleção estão em negrito.

pela correlação de Pearson obtiveram em média melhores resultados que as segmentações utilizando os modelos analisados. Deve-se ressaltar que os canais selecionados pela correlação não foram os mesmos para todas as cinco segmentações. Isto é justificado devido ao fato da análise das informações de cores serem feitas nos conjuntos de pixels sementes, pois cada usuário selecionou um conjunto diferente de sementes.

As melhores segmentações foram obtidas utilizando o conjunto de sementes selecionado pelo Usuário 4. Nessa segmentação foram rotulados corretamente 95,67% dos pixels com a utilização do canal $I_1I_2I_3$ e 95,18% com a utilização dos canais III_3 (ver Figura 5.18). Observando a Tabela 5.5 pode-se observar um *outlayer* na segmentação do Usuário 1 utilizando o modelo *HSI*. Isso foi provocado pela adição de pixels sementes no objeto computador que ficaram na borda entre o computador e parede fazendo que muitos pixels da parede fossem rotulados como computador.

Já os resultados apresentados pela seqüência do *Office* com adição de ruído gaussiano, a média dos resultados apresentados com a utilização do modelo de cor $I_1I_2I_3$ foram ligeiramente melhores que os resultados obtidos pelos canais selecionado pela correlação de Pearson, 91,91% contra 91,48%. Mas em compensação, três dos cinco usuários conseguiram melhores segmentações utilizando os canais obtidos pela correlação. No-

	<i>RGB</i>	<i>HSI</i>	$I_1I_2I_3$	<i>Corr</i>
Usuário 1	84,18 ± 0,64	92,13 ± 0,78	94,05 ± 0,41	94,75 ± 0,42
Usuário 2	86,79 ± 0,47	89,76 ± 0,33	89,97 ± 0,31	90,97 ± 0,40
Usuário 3	85,34 ± 0,47	90,16 ± 0,43	88,89 ± 0,21	92,02 ± 0,36
Usuário 4	86,78 ± 0,44	92,12 ± 1,25	93,96 ± 0,67	92,28 ± 0,61
Usuário 5	87,00 ± 0,43	90,10 ± 0,40	92,67 ± 0,33	87,38 ± 0,47
Média	86,02 ± 0,33	90,85 ± 0,64	91,91 ± 0,39	91,48 ± 0,48

Tabela 5.6: Resultados das segmentações do vídeo sintético com a adição de ruídos, onde os melhores resultados para cada seleção estão em negrito.

vamente as melhores segmentação foram obtidas com utilização dos canais selecionados pela correlação (I_2I_1S) e pelo modelo $I_1I_2I_3$, respectivamente 94,75% e 94,05% de pixels classificados corretamente (ver Figura 5.19). A média dos pixels rotulados corretamente para a seqüência do *Office* foi de aproximadamente de 90,78% para a seqüência original e 90,06% para a seqüência com ruído. Isso mostra que o algoritmo de segmentação fuzzy possui uma boa tolerância na segmentação de vídeos ruidosos.

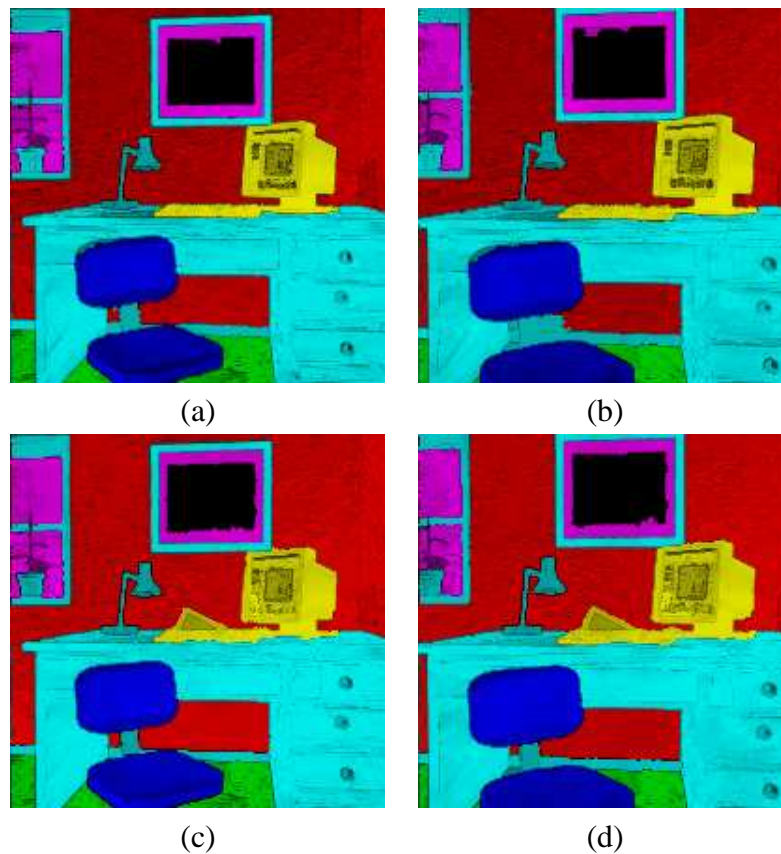


Figura 5.18: Segmentação da seqüência original do *Office*. Frame 5 e 15 segmentados utilizando o modelo $I_1I_2I_3$ (a-b) e com os canais HII_3 (c-d).

Para a seqüência do *Demost*, a configuração que produziu a melhor segmentação foi a que utilizou 30% de informação de movimento (lambda igual a 800, 4 níveis e 15 iterações) e 70% de informação de cor (modelos selecionados pela correlação). Depois da primeira segmentação, foram adicionadas novas sementes com a intenção de melhorar a segmentação produzida pelo modelos de cor *RGB*, até então o que tinha obtido a pior segmentação. Depois de dois ciclos de segmentação, ou seja, adição ou remoção de sementes e nova segmentação, os resultado obtidos ficaram bem próximos dos obtidos com os canais selecionados pela correlação. A Figura 5.20 apresenta 4 frames da segmentação da seqüência *Demost* usando o modelo *RGB*.

A última seqüência analisada, a *Caneca*, a configuração que produziu o melhor resul-

tado foi com 20% pra informação e 80% para informação de cor. Os parâmetros utilizados para gerar a estimação de movimento: lambda 400, 4 níveis e 12 iterações. O modelo de cor $I_1I_2I_3$ foi o que gerou a melhor segmentação para esta seqüência. A Figura 5.21 contém os resultados das segmentações desta seqüência.

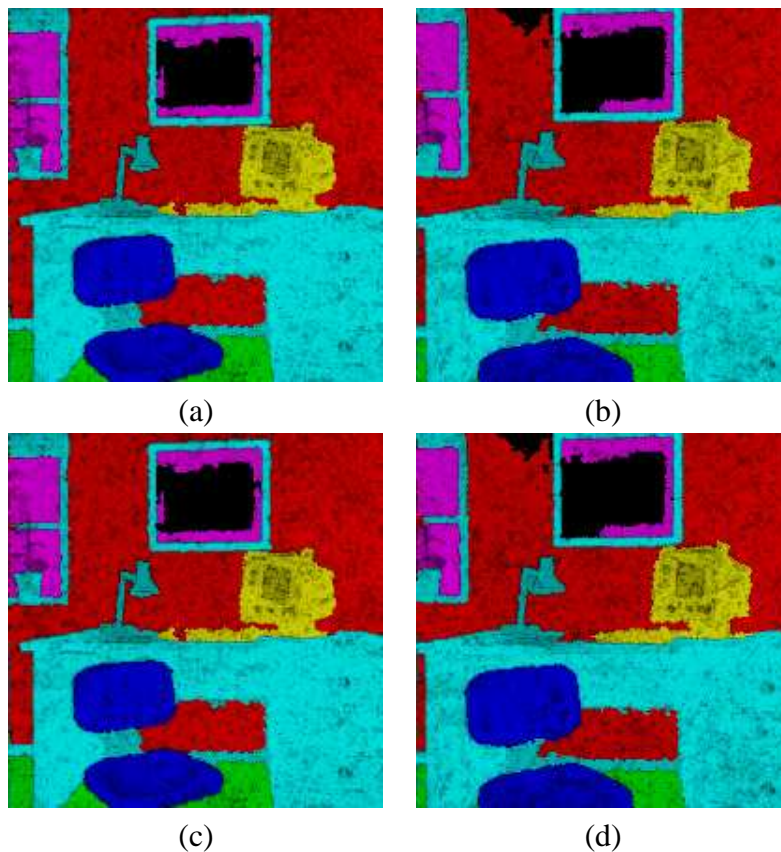


Figura 5.19: Segmentação da seqüência *Office* com ruído. Frame 5 e 15 segmentados utilizando o modelo $I_1I_2I_3$ (a-b) e com os canais I_2I_1S (c-d).

A Tabela 5.7 mostra o tempo de segmentação para os seis vídeos analisados nesse experimento. Nas condições apresentadas nesta subseção, o tempo médio para classificar um pixel é de aproximadamente $0,4 \mu m.s.$

Seqüência	Qtd. Frames	Qtd. Pixel	Tempo
Traffic Car	71	320 X 240	21 seg
Office	20	200 X 200	3 seg
Demost	21	360 X 240	7 seg
Taz	75	360 X 240	26 seg
Caneca	75	360 X 240	26 seg
Sapo	85	360 X 240	30 seg

Tabela 5.7: Tempo da segmentação fuzzy para os vídeos analisados.

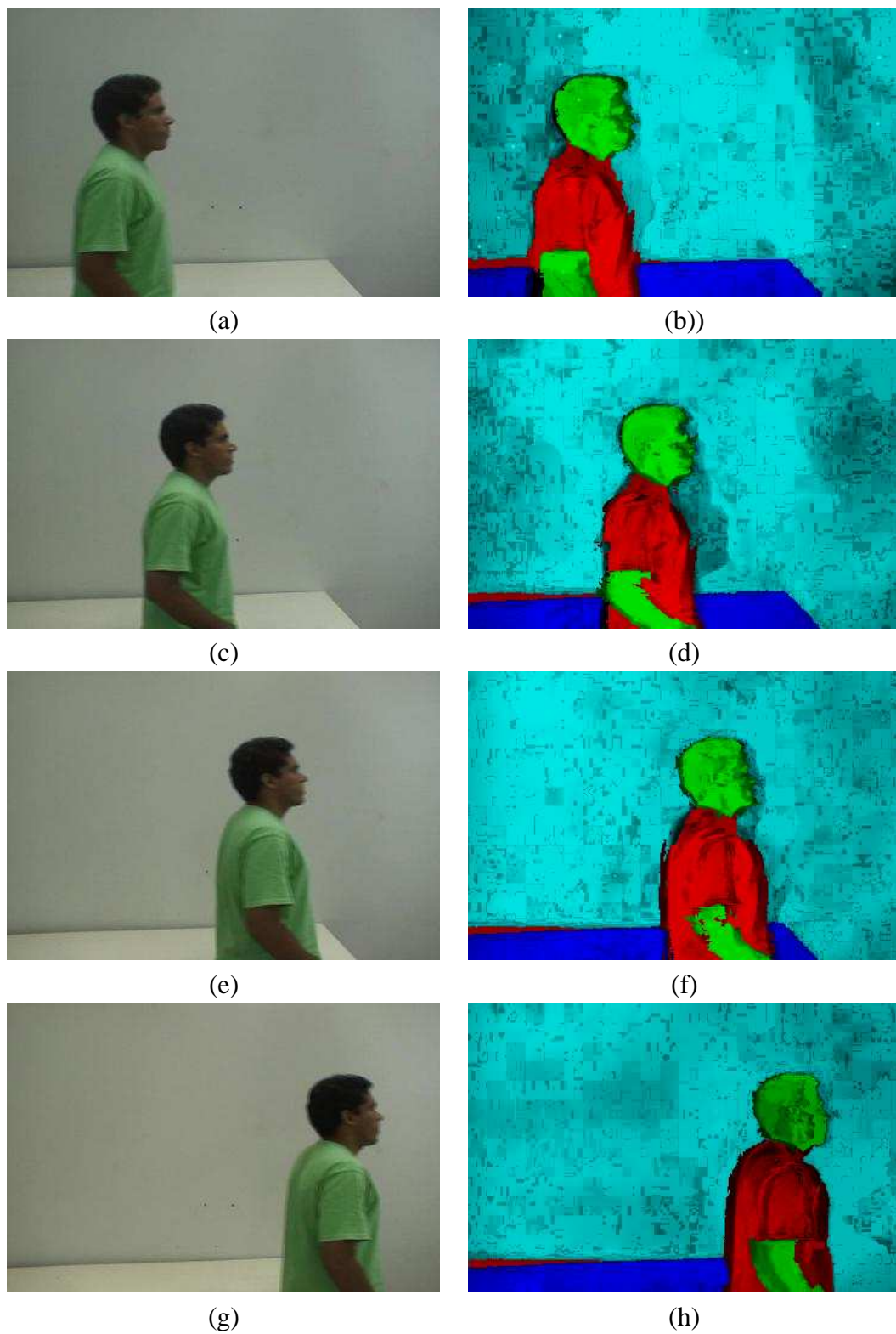


Figura 5.20: Resultado da segmentação da seqüência do *Demost*. Os frames 1, 7, 14 e 21 originais (a,c,e,g) e segmentados com o modelo *RGB* (b-d-f-h).

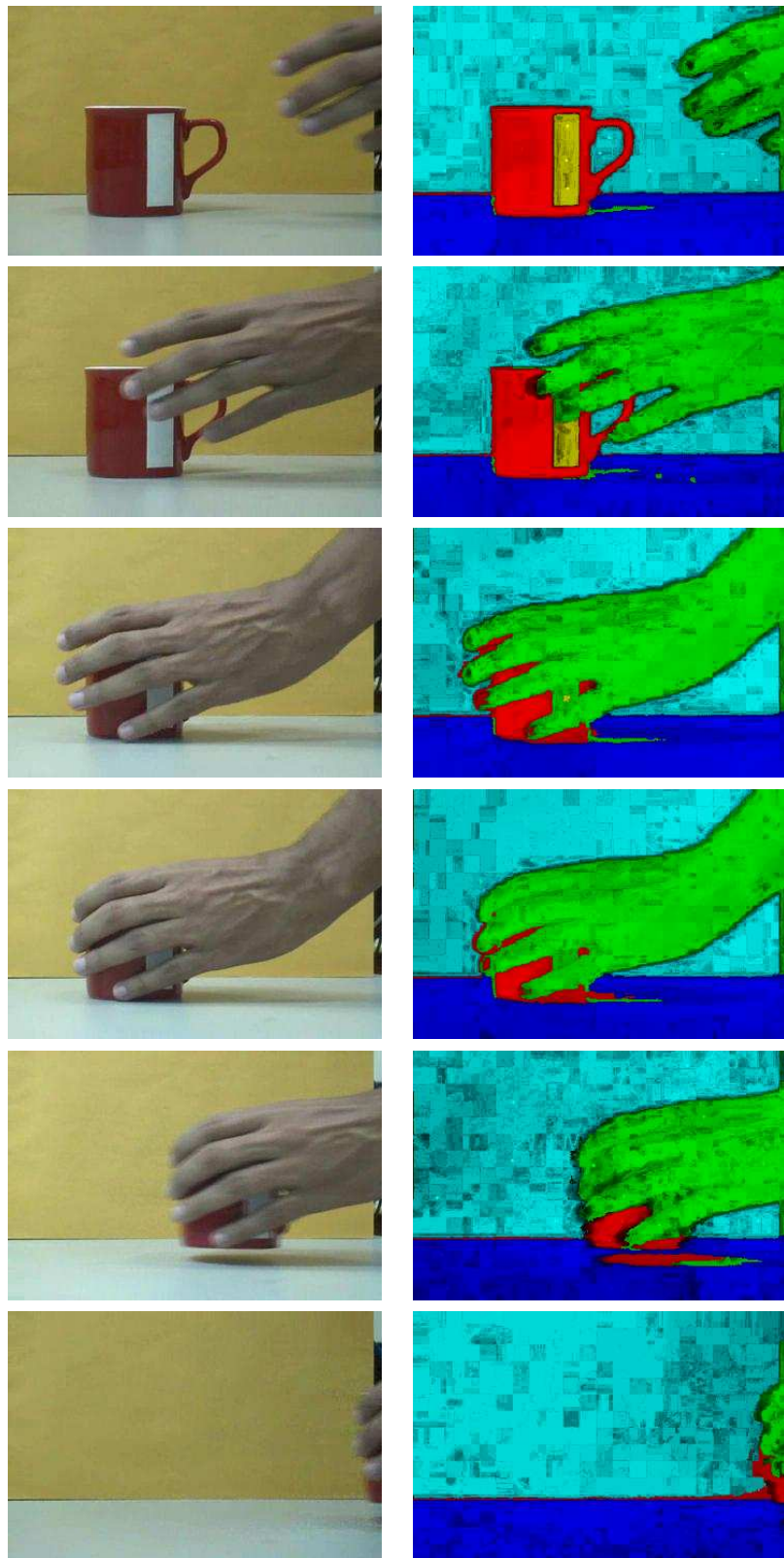


Figura 5.21: Resultado da segmentação da seqüência da *Caneca* com o modelo de cor $I_1I_2I_3$. Cada linha corresponde a um intervalo de 15 frames.

5.6 Uma Aplicação da segmentação Fuzzy para renderizações NPR

Renderização não realista pode ser definida como o processamento de imagens e vídeos sob uma perspectiva de arte, onde as imagens em vídeos geradas nesse processo simulam técnicas artísticas, tentando expressar visualmente características do estilo artístico utilizado [CBNO06]. A utilização de técnicas de renderização não foto realísticas para estilização de vídeos foi introduzida em Litwinowicz [Lit97], sendo essas técnicas geralmente divididas em duas etapas distintas: a segmentação do vídeo para extração dos objetos de interesse e a renderização das regiões de interesse. Alguns exemplos de renderizações não realista para estilização de vídeos podem ser vistos em Wang et al. [WXSC04], na criação de desenhos animados de vários estilos, em Hertzmann e Perlin [HP00], na pintura em tela, em Smith et al. [SLK05], com animação em mosaicos, e em Carvalho et al. [CBNO06] com vídeos em garrafas com areia.

A seqüência que será utilizada para demonstrar a utilização da segmentação fuzzy na renderização não realista, foi produzida pela equipe do projeto AnimVídeo e possui 85 frames. Foram utilizadas duas técnicas para fazer as renderizações: *Csand* e pintura impressionista. Para esse exemplo, foram segmentados 3 objetos: background da cena; corpo do sapo; e barriga e joelho do sapo. A Figura 5.22 ((a) e (b)) mostra um frame original e a sua segmentação fuzzy respectivamente. Nessas duas técnicas, a segmentação fuzzy foi utilizada para possibilitar a aplicação da renderização em apenas alguns objetos da cena.

A primeira técnica, denominada *Csand* [CBNO06], está sendo desenvolvida pela equipe do projeto AnimVídeo e tenta reproduzir o mesmo efeito aplicado por artistas que criam desenhos dentro de garrafas com areias de diferentes cores. Nesses efeitos são analisados os pixels ou região dos frames do vídeo a ser renderizado, e então são aplicadas nos frames texturas procedurais que foram geradas por uma função de ruído gaussiano [CBNO06]. A Figura 5.22 mostra dois frames da renderização NPR *Csand* para a seqüência do sapo. Devido a utilização do mapa de segmentação, podemos aplicar diferentes estilos de renderização nos objetos que foram segmentados.

A segunda renderização NPR apresentada aqui é uma técnica de renderização que simula a pintura impressionista. Essa técnica foi implementada pela equipe da AnimVídeo e utiliza a magnitude e a orientação do fluxo óptico para determinar a orientação e o comprimento de cada pincelada [GdSdC06]. Nesse método, o mapa da segmentação obtido pela segmentação fuzzy é utilizado tanto para a separação dos objetos como para auxiliar o cálculo do fluxo óptico de um objeto específico. A Figura 5.23 mostra três frames da renderização com pintura impressionista na seqüência do sapo.

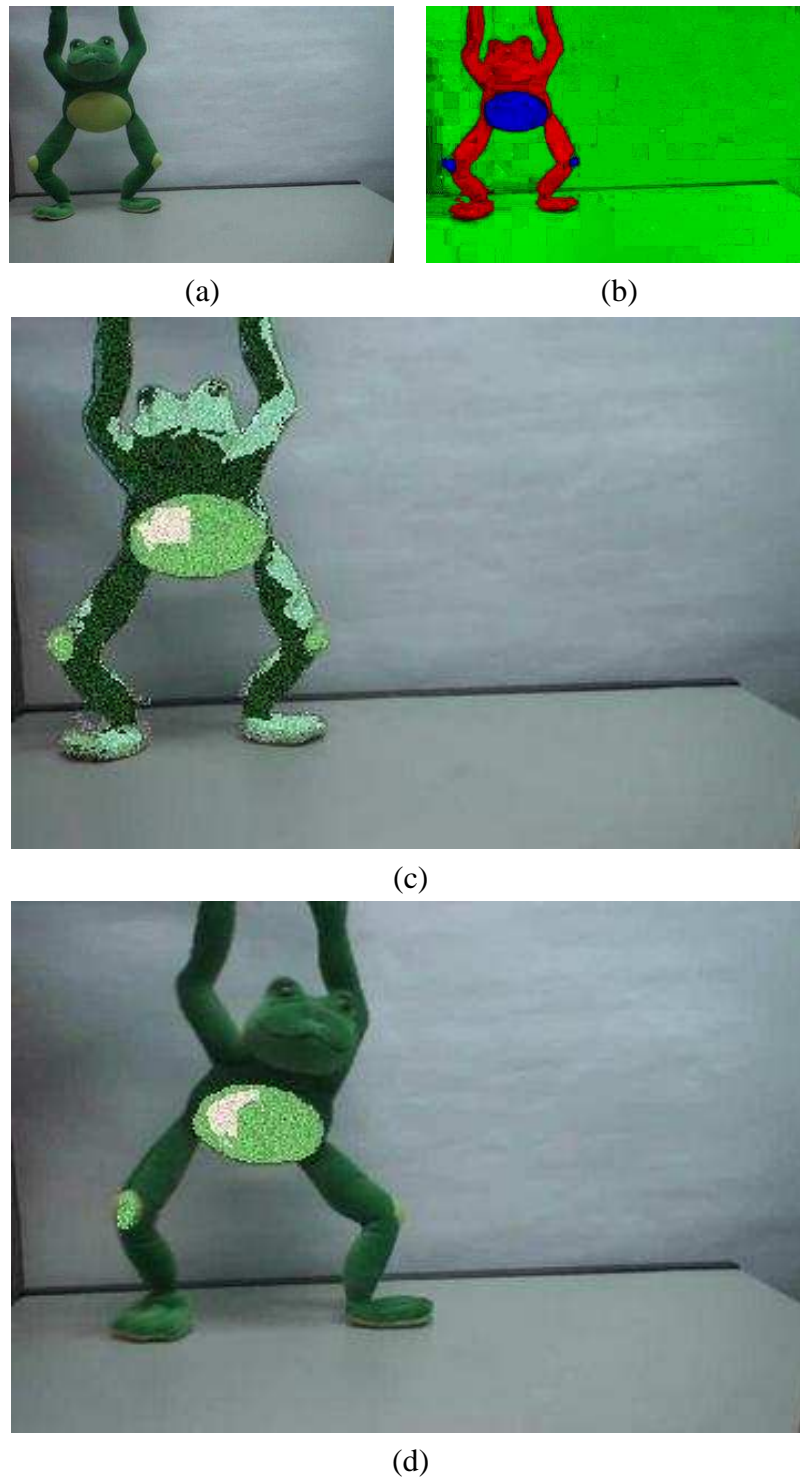


Figura 5.22: Frame original (a) e o mapa de segmentação (b) do primeiro frame, e a Renderização NPR *Csand* em dois frames da seqüência. O primeiro renderizando só o sapo (c) e o segundo só a barriga e joelho do sapo (d).



(a)



(b)



(c)

Figura 5.23: Exemplo da aplicação da segmentação fuzzy para renderização NPR. Três frames utilizando a técnica de pintura impressionista.

Capítulo 6

Conclusão

Com a intenção de superar as dificuldades encontradas para segmentar imagens e vídeos corrompidos por ruídos, e gerar segmentações para serem utilizadas no processo de renderização não realista, o presente trabalho realizou uma investigação dos algoritmos de segmentação fuzzy desenvolvidos por Herman e Carvalho [HC01, CHK05] e propôs uma extensão desse algoritmos para segmentar vídeos.

Para verificar a robustez e a acurácia do algoritmo de segmentação fuzzy MOFS, foi desenvolvida uma metodologia para identificar e selecionar partículas em micrografos gerados pela Microscopia Eletrônica. Devido ao processo utilizado para aquisição desses micrografos, as imagens geradas possuem uma alta taxa de ruído. A utilização da segmentação fuzzy nessa metodologia mostrou-se contribuiu significativamente para redução da taxa de Falsos Positivos e Falsos Negativos.

A extensão do algoritmo de segmentação fuzzy fast-MOFS para segmentar vídeos coloridos apresentado nesse trabalho mostrou bons resultados tanto na segmentação de vídeos reais como sintéticos. Nessa extensão, o vídeo é tratado como um volume 3D em que o usuário pode indicar os pixels sementes de cada objeto em qualquer frame dos vídeos, e então as funções de afinidade calcularão a probabilidade de um pixel pertencer aos objetos segmentados. Essas funções de afinidade utilizarão informações de cor e/ou movimento para rotular os pixels ao objeto mais adequado.

As informações de cores utilizadas nas funções de afinidades podem ser fornecidas por um modelo de cores ou por uma metodologia, aqui apresentada, que se propõe selecionar três canais de cores dentre os canais de um conjunto de modelos de cores. Para essa seleção foi utilizada a correlação de Pearson com a intenção de selecionar as informações de cores menos correlacionadas. Já as informações de movimento foram obtidas utilizando estimativa de movimento por fluxo óptico, através de uma versão multi-resolução do algoritmo de Proesmans [PGPO94] desenvolvido por McCane [MNCG01].

Para segmentar as seqüências de vídeos foi desenvolvida uma ferramenta que permite ao usuário segmentar interativamente vídeos através da adição e remoção de sementes para cada objeto a ser segmentado. Utilizando essa ferramenta o usuário poderá também fazer a estimação de movimento através do fluxo óptico, escolher o modelo ou os canais de cores para gerar o volume a ser segmentado, e por fim segmentar e visualizar o resultado da segmentação.

Os resultados das segmentações obtidas nos experimentos, em sua maioria, mostraram-se robustos e consistentes ao longo dos frames segmentados. A utilização de uma interface gráfica para seleção das sementes contribuiu significativamente para o sucesso da segmentação. Através dela o usuário pode selecionar as sementes mais apropriadas para fornecer informações à função de afinidades. O tempo de segmentação, em torno de $0,4 \mu s$ por pixel, é bastante rápido e possibilita o usuário interagir com a ferramenta para adicionar e remover sementes e assim obter um melhor resultado na segmentação.

O método de segmentação apresentou-se bastante robusto na segmentação sobre influência de ruídos. Para a seqüência sintética aqui apresentada, os resultados apresentados para a seqüência original, 90,78% de pixels rotulados corretamente, e para com seqüência adição de ruídos, 90,06%, foram bastante próximos, mostrando assim que a função de afinidade fuzzy aqui utilizada é robusta a presença desse tipo de ruído.

A metodologia apresentada para seleção de canais de cores, apesar de não garantir a produção das melhores segmentações em comparação com os modelos de cores aqui utilizados, obteve, quando não melhores, resultados muito próximos das melhores segmentações com um dos modelos utilizados. Em nenhuma das seqüências analisadas a segmentação com a utilização dos canais selecionados com a correlação tiveram os piores resultados. A utilização da metodologia de seleção pela correlação de Pearson pode ser justificativa em situações em que não seja conhecido o melhor modelo de cor para segmentar determinada seqüência e/ou quando só é possível segmentar a seqüência uma única vez.

Como trabalho futuro pode ser adicionado ao algoritmo de segmentação um mecanismo para se verificar a confiabilidade da estimação de movimento obtida através do fluxo óptico. Com essa medida de confiabilidade o programa poderia sugerir ou atribuir automaticamente peso a informação de movimento na composição de função de afinidade. Para melhorar o uso de informação de movimento nas funções de afinidade, poderia ser analisada a utilização das informações de movimento somente dos pixels sementes mais próximos do pixel segmentado. Atualmente é utilizada a informação de movimento de todos os pixels semente do objeto analisado. Com isso poderia se evitar imprecisão na segmentação de objetos que possuem movimentos inconstantes (rápidos em um trecho e lento em outro) no decorrer do vídeo.

Uma outra sugestão seria a utilização de uma medida de compactação e dispersão, como a apresentada em VandenBroucke [VMP98], com a intenção de se atribuir pesos as informações de cada canal de um modelo de cor utilizado para segmentar. Com isto os canais que mais separasse os pixels de diferentes objetos e que mais compactasse os pixels que pertencesse ao mesmo objeto ganhariam um maior peso nas funções de afinidade fuzzy.

Referências Bibliográficas

- [BA90] M.J. Black and P. Anandan. A model for the detection of motion over time. In *Third International Conference on Computer Vision*, volume 90, pages 33–37, Osaka, Japan, dec 1990.
- [BB95] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computer Surveys*, 27(3):433–466, September 1995.
- [Bev02] Alessandro Bevilacqua. Effective object segmentation in a traffic monitoring application. In *Third Indian Conference on Computer Vision, Graphics & Image Processing*, pages 125–130, 2002.
- [BF80] S.T. Barnard and M.A. Fischler. Computational and biological models of stereo vision. *DARPA Image Understanding Workshop*, 90:439–448, September 1980.
- [Bie88a] J. Biemond. Pel-recursive wiener-based algorithm for the simultaneous estimation of rotation and translation. In *Proc. SPIE Visual Communications and Image Processing '88: Third in a Series*, volume 1001, pages 917–924, Cambridge, MA, November 1988.
- [Bie88b] M. Bierling. Displacement estimation by hierarchical block matching. In *Proc. SPIE Visual Communications and Image Processing*, volume 1001, pages 942–950, Boston, November 1988.
- [Bov05] A. C. Bovik, editor. *Handbook of Image and Video Processing*. Academic Press, second edition, 2005.
- [BT05] J.L. Barron and N.A. Thacker. Tutorial: Computing 2d and 3d optical flow. Technical Report 2004-012, University of Manchester, Medical School, Imaging Science and Biomedical Engineering Division, january 2005.
- [CBNO06] B.M. Carvalho, L.S. Britto Neto, and L.M. Oliveira. Bottled sand movies. In *CGIV '06: Third International Conference on Computer Graphics, Imaging and Visualisation*, pages 402–407, Sydney, Austrália, July 2006. IEEE.
- [CGHK99] B.M. Carvalho, C.J. Gau, G.T. Herman, and T.Y. Kong. Algorithms for fuzzy segmentation. *Pattern Analysis Application*, 2(1):73–81, April 1999.

- [Che93] C.-T Chen. Video compression: standards and applications. *Journal Communication and Image Sentation*, 4(2):103–111, 1993.
- [CHK05] Bruno M. Carvalho, Gabor T. Herman, and T. Yung Kong. Simultaneous fuzzy segmentation of multiple objects. *Discrete Applied Mathematics*, 151(1-3):55–77, 2005.
- [CIE31] CIE. *CIE - Commission Internationale de l'Eclairage*. Cambridge University Press, Cambridge, 1931.
- [CJSW01] H.D. Cheng, H. Jiang, Y. Sun, and J.I. Wang. Color image segmentation: Advances and prospects. *Pattern Recognition*, 34(12):2259–2281, 2001.
- [CL94] T. Carron and P. Lambert. Color edge detector using jointly hue, saturation and intensity. In *International Conference on Image Processing*, volume 3, pages 977–981, Austin, TX, USA, November 1994. IEEE.
- [CLR90] T. H. Cormen, C.E Leiserson, and R.L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.
- [CM97] D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 750, Washington, DC, USA, 1997. IEEE Computer Society.
- [CNO06] B. M. Carvalho, L.S. Brito Neto, and L. M. Oliveira. Bottled sand movies. In *International Conference on Computer Graphics, Imaging and Visualization (CGIV'06)*, pages 402–407, Sidney, Australia, 2006. IEEE CS Press.
- [COA06] B. M. Carvalho, L. M Oliveira, and G. S. Andrade. Fuzzy segmentation of color video shots. In *13th International Conference on Discrete Geometry for Computer Imagery (LNCS)*, volume 4245, pages 494–500, Berlin, 2006. Springer.
- [COG06] B M Carvalho, L M Oliveira, and E Garduno. Semi-automatic single particle segmentation on electron micrographs. In *International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1024–1027, Washington, DC, USA, April 2006. IEEE.
- [CVR07] CVRG. Optical flow. University of Otago. Department of Computer Science. Computer Vision Research Group <http://www.cs.otago.ac.nz/research/vision/Research/OpticalFlow/opticalflow.html>, janeiro 2007.
- [ESE93] Y.M. Erkan, M.I. Sezan, and A.T Erdem. A hierarchical phase-correlation method for motion estimation. In *Conference on Information Sciences and Systems*, pages 419–424, Baltimore, MD, March 1993.

- [FHPZ96] C. Firmin, D. Hamad, J. Postaire, and R.D. Zhang. Feature extraction and selection for fault detection in production of glass bottles. *Machine Graphics Vision International Journal*, 5(1):77–86, 1996.
- [FWM⁺02] J. Frank, T. Wagenknecht, B.F. McEwen, M. Marko, C.-E. Hsieh, and C. A. Mannella. Three-dimensional imaging of biological complexity. *Journal of Structural Biology*, 138(1-2):85–91, 2002.
- [GdSdC06] Rafael Gomes, Tiago Souza dos Santos, and Bruno Motta de Carvalho. Coerência temporal intra-objeto para npr utilizando fluxo Ótico restrito. Anais do Simpósio Brasileiro de Computação Gráfica, Processamento de Imagens e Visão Computacional, outubro 2006.
- [GH92] J. Gauch and C. Hsia. A comparison of three color image segmentation algorithm in four color space. In *SPIE Visual Communications and Image Processing '92*, volume 1818, pages 1168–1181, 1992.
- [GIM07] Gimp - gnu image manipulation program. <http://www.gimp.org/>, fevereiro 2007.
- [GL00] S. Galic and S. Loncaric. Spatio-temporal image segmentation using optical flow and clustering algorithm. In *First International Workshop on Image and Signal Processing and Analysis*, pages 63–68, Pula, Croatia, 2000.
- [GM90] H. Gharavi and M. Mills. Block matching motion estimation algorithms: New results. *IEEE Transactions on Circuits and Systems*, 37(5):649–651, May 1990.
- [GMN⁺98] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: An evaluation of eight optical flow algorithms. In *Proceedings of Ninth British Machine Vision Conference*, pages 195–204, Southampton, United Kingdom, 1998.
- [GS96] T. Gevers and A. Smeulders. A comparative study of several color models for color image invariant retrieval. In *First Int'l Workshop Image Databases and Multimedia Search*, pages 7–27, 1996.
- [GT01] I. Grinias and G. Tziritas. A semi-automatic seeded region growing algorithm for video object localization and tracking. *Signal Processing: Image Communication*, 16:977–986, Aug 2001.
- [GW02] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice Hall, second edition, 2002.
- [Hai05] J. F Hair. *Análise Multivariada de Dados*. Bookman, 5th edition, 2005.
- [HC01] Gabor T. Herman and Bruno M. Carvalho. Multiseeded segmentation using fuzzy connectedness. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 23(5):460–474, 2001.
- [Her98] G.T Herman. *Geometry of Digital Spaces*. Springer, 1998.

- [HP00] Aaron Hertzmann and Ken Perlin. Painterly rendering for video and interaction. In *NPAR 00: Proceedings of the 1st International Symposium on Non-photorealistic Animation and Rendering*, pages 7–12, New York, NY, USA, 2000. ACM Press.
- [HS81] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
- [HS99] W. L. Hsu and C. W. Shih. Video segmentation based on watershed algorithm and optical flow motion estimation. In *IV Multimedia Storage and Archiving Systems*, volume 3846, pages 56–66, Sethuraman Panchanathan, August 1999.
- [JMF99] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computer Surveys*, 31(3):264–323, 1999.
- [JW87] Lowell Jacobson and Harry Wechsler. Derivation of optical flow using a spatiotemporal-frequency approach. *Computer Vision, Graphics and Image Processing*, 38(1):29–65, 1987.
- [KK03] Jong Bae Kim and Hang Joon Kim. Efficient region-based motion segmentation for a video monitoring system. *Pattern Recogn. Lett.*, 24(1-3):113–128, 2003.
- [KS01] S. Khan and M. Shah. Object based segmentation of video using color motion and spatial information. In *Computer Vision and Pattern Recognition*, volume 2, pages 746–751. IEEE, 2001.
- [Lit97] P. Litwinowicz. Processing images and video for an impressionist effect. In *ACM SIGGRAPH*, pages 407–414, 1997.
- [LK81] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
- [LL06] C. Lodato and S. Lopes. An optical flow based segmentation method for objects extraction. *Transactions on Engineering, Computing and Technology*, 12:41–46, March 2006.
- [LP80] H.C. Longuet and K. Prazdny. The interpretation of a moving retinal image. *Royal Society of London*, 208:385–397, August 1980.
- [LY94] J. Liu and Y.-H. Yang. Multiresolution color image segmentation. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 16(3):689–700, 1994.
- [MNCG01] B. McCane, K. Novins, D. Crannitch, and B. Galvin. On benchmarking optical flow. *Computer Vision and Image Understanding*, 84:126–143, 2001.
- [OKS80] Y. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13(3):222–241, July 1980.

- [OPC⁺06] L. M. Oliveira, R. B. Paradedda, B.M. Carvalho, G.S. Andrade, and A. M. P. Canuto. Segmentação fuzzy de sequências de vídeo utilizando espaços de cores híbridos. In *II Workshop de Visão Computacional*, volume 1, pages 217–222, São Carlos, Brasil, October 2006. São Carlos : EECS-USP.
- [PDM86] Sankar K. Pal and Dwijesh K. Dutta-Majumder. *Fuzzy mathematical approach to pattern recognition*. Halsted Press, New York, NY, USA, 1986.
- [PGPO94] Marc Proesmans, Luc J. Van Gool, Eric J. Pauwels, and Andr#233; Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. In *ECCV '94: Proceedings of the Third European Conference-Volume II on Computer Vision*, pages 295–304, London, UK, 1994. Springer-Verlag.
- [PP93] N.R. Pal and S.K. Pal. A review on image segmentation techniques. *Pattern Recognition Society*, 26:1277–1294, 1993.
- [PW04] F. Porikli and Y. Wang. Automatic video object segmentation using volume growing and hierarchical clustering. *EURASIP Journal on Applied Signal Processing*, 3(2):442–453, March 2004.
- [RF04] B.K. Rath and J. Frank. Fast automatic particle picking from cryo-electron micrographs using a locally normalized cross-correlation. *Journal of Structural Biology*, 145:84–90, 2004.
- [Ros79] A. Rosenfeld. Fuzzy digital topology. *Information and Control*, 40(1):76–87, 1979.
- [Ros03] A.M. Roseman. Particle finding in electron micrographs using fast local correlation algorithm. *Ultramicrosc.*, 94:225–236, 2003.
- [Sin91] A. Singh. *Optic Flow Computation: A Unified Perspective*. IEEE Computer Society Press, Los Alamitos, California, 1991.
- [SJ06] M. Summerfield and J.Blanchette. *C++ GUI Programming with QT 4*. Prentice Hall, 2006.
- [SLK05] Kaleigh Smith, Yunjun Liu, and Allison Klein. Animosaiacs. In *SCA 05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 201–208, New York, NY, USA, 2005. ACM Press.
- [TDA98] J. C. Terrillon, M. Davis, and S. Akamatsu. Detection of human faces in complex scene images by use of a skin color modelo and of invariant fourier-mellin momento. In *IEEE International Conference on Pattern Recognition*, volume 2, pages 1350–1355, Brisbane - Australia, August 1998.
- [Tek95] A.M. Tekalp. *Digital Video Processing*. Prentice Hall, 1995.
- [Tek05] A.M. Tekalp. *Handbook of Image and Video Processing*, chapter 4, pages 771–791. Academic Press, second edition, 2005.

- [Tro07] Trolltech. Qt cross-platform c++ development. <http://www.trolltech.com/products/qt>, janeiro 2007.
- [US96] J.K. Udupa and S Samarasekera. Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation. *Graphical Models and Image Processing*, 58:246–261, 1996.
- [VCPB03] D. L. Vilariño, D. Cabello, X.M. Pardo, and V.M. Brea. Video segmentation for traffic monitoring tasks based on pixel-level snakes. In *1st Iberian Conference on Pattern Recognition and Image Analysis*, volume 2652, pages 1074–1081, Puerto de Andratx, Espanha, 2003. Lectures Notes in Computer Science.
- [vHGM⁺00] M. van Heel, B. Gowen, R. Matadeen, E.V. Orlova, R. Finn, T. Pape, D. Cohen, H. Stark, R. Schmidt, M. Schatz, and A. Patwardhan. Single-particle electron cryo-microscopy: towards atomic resolution. *Quarterly Review Biophysics*, 33(4):307–69, November 2000.
- [VMP98] N. VandenBroucke, L. Macaire, and J.G. Posaire. Color pixels classification in an hybrid color space. In *IEEE Conference on Image Processing*, pages 176–180. IEEE, 1998.
- [VMP03] Nicolas Vandenbroucke, Ludovic Macaire, and Jack-Géard Postaire. Color image segmentation by pixel classification in an adapted hybrid color space: application to soccer image analysis. *Comput. Vis. Image Underst.*, 90(2):190–216, 2003.
- [VP89] A. Verri and T. Poggio. Motion field and optical flow: Qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):490–498, 1989.
- [WOZ02] Y. Wang, J. Ostermann, and Y Zhang. *Video Processing and Communications*. Prentice Hall, 2002.
- [WS82] G. Wyszecki and W.S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley, NewYork, 1982.
- [WXSC04] Jue Wang, Yingqing Xu, Heung-Yeung Shum, and Michael F. Cohen. Video tooning. *ACM Transactions Graphics*, 23(3):574–583, 2004.
- [ZC97] David X. Zhong and S.F Chang. Video object model and segmentation for content-based video indexing. In *IEEE International Conference on Circuit and Systems*, pages 1492–1495, Hong Kong, 1997. IEEE.
- [ZCG⁺04] Y. Zhu, B. Carragher, R.M. Glaeser, C. Bajaj D. Fellmann, M. Bern, F. Mouche, F. de Haas, R. J. Hall, D. J. Kriegman S. J. Ludtke, S.P. Mallick, P. A. Penzeck, A.M. Roseman, F. J. Sigworth, N. Volkman, and C.S. Potter. Automatic particle selection: Results of a comparative study. *Journal of Structural Biology*, 145:3–14, 2004.

-
- [ZMSH03] J. P. Zubelli, R. Marabini, C. O. S. Sorzano, and G. T. Herman. Three-dimensional reconstruction by chahine's method from electron microscopic projections corrupted by instrumental aberrations. *Inverse problems*, 19(4):933–949, 2003.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)